

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.Doi Number

# Building a Scalable and Interpretable Bayesian Deep Learning Framework for Quality Control of Free Form Surfaces

**Sumit Sinha<sup>1</sup>, Pasquale Franciosa<sup>1</sup>, and Dariusz Ceglarek<sup>1</sup>**

<sup>1</sup>WMG, University of Warwick, Coventry (CV4 7AL), United Kingdom (U.K.)

Corresponding author: Sumit Sinha (e-mail: [sumit.sinha.1@warwick.ac.uk](mailto:sumit.sinha.1@warwick.ac.uk)).

This work was supported by the UK EPSRC project EP/K019368/1: “Self-Resilient Reconfigurable Assembly Systems with In-process Quality Improvement” and the WMG-IIT scholarship.

**ABSTRACT** Deep learning has demonstrated high accuracy for 3D object shape error modeling necessary to estimate dimensional and geometric quality defects in multi-station assembly systems (MAS). Increasingly, deep learning-driven Root Cause Analysis (RCA) is used for decision-making when planning corrective action of quality defects. However, given the current absence of scalability enabling models, training deep learning models for each individual MAS is exceedingly time-consuming as it requires large amounts of labelled data and multiple computational cycles. Additionally, understanding and interpreting how deep learning produces final predictions while quantifying various uncertainties also remains a fundamental challenge. In an effort to address these gaps, a novel closed-loop in-process (CLIP) diagnostic framework underpinned algorithm portfolio is proposed which simultaneously enhances scalability and interpretability of the current Bayesian deep learning approach, Object Shape Error Response (OSER), to isolate root cause(s) of quality defects in MAS. The OSER-MAS leverages a Bayesian 3D U-Net architecture integrated with Computer-Aided Engineering simulations to estimate root causes. The CLIP diagnostic framework shortens OSER-MAS model training time by developing: (i) closed-loop training to enable faster convergence for a single MAS by leveraging uncertainty estimates of the Bayesian 3D U-net model; and, (ii) transfer/continual learning-based scalability model to transmit meta-knowledge from the trained model to a new MAS resulting in convergence using comparatively less training samples. Additionally, CLIP increases the transparency for quality-related root cause predictions by developing interpretability model which is based on 3D Gradient-based Class Activation Maps (3D Grad-CAMs) and entails: (a) linking elements of MAS model with functional elements of the U-Net architecture; and, (b) relating features extracted by the architecture with elements of the MAS model and further with the object shape error patterns for root cause(s) that occur in MAS. Benchmarking studies are conducted using six automotive-MAS with varying complexities. Results highlight a reduction in training samples of up to 56% with a loss in performance of up to 2.1%.

**INDEX TERMS** Bayesian Deep Learning, Continual Learning, Transfer Learning, Multi-station Assembly, U-Net, 3D Convolutional Neural Networks (CNN), Manufacturing

## I. INTRODUCTION

### A. PROBLEM DESCRIPTION

The manufacturing industry is currently undergoing a data revolution [1][2] which requires scalable and interpretable approaches that enable digitalization and integration of in-process monitoring data (i.e. 3D scanners), Artificial Intelligence (AI) algorithms [3], and Computer-Aided

Engineering (CAE) simulations to support current Industry 4.0 strategies such as Zero-Defect-Manufacturing and Right-First-Time. These strategies collectively drive towards scenarios where all quality requirements of manufactured products are satisfied starting from Job 1 (right-first-time strategy) and continue during the lifecycle of the product (near zero-defect manufacturing strategy). This is a very challenging task due to increased product variety and smaller

batch sizes with ever-decreasing time-to-market. Currently, numerous modern manufacturing systems applications use multiple robotic assembly stations placed in a production line to improve productivity and product quality. For example, a typical automotive body assembly (Body-in-White (BIW)) consists of hundreds of stamping parts and components processed along 60-85 assembly stations with a production rate of 30-65 vehicles per hour with 3-5 varieties of vehicles being produced simultaneously on a single assembly line. Automatic robotic assembly lines can significantly increase productivity and the variety of products produced on a single line. Nonetheless, the embedded complexity of an automatic production line can lead to failures involving robot, end effector with joining head, and/or fixtures, resulting in diminished product quality, scrap, rework, or production downtimes. As a result, there is an urgent need for diagnostics of modern automatic assembly lines. The diagnostics approaches should have the ability to work with current requirements of modern assembly lines, i.e., have capability for (i) multi-variety products with non-rigid parts and shape variances which are rapidly scaled-up from low to high volume production (this translates to the need for rapid *scalability* of diagnostics models between products and production volumes); and (ii) automatic recovery from disruptions at minimum cost (translating to need for *interpretability* of the diagnostic results, which will provide transparency and the causes of their prediction, as they are often used to plan costly corrective action of quality defects).

Therefore, it is crucial to develop diagnostic approaches that are scalable and interpretable to ensure that multi-station assembly systems (MAS) can continue to produce high-quality products in the presence of uncertainties induced by non-ideal parts and operational errors [1][4]. Recently Object Shape Error Response (OSER) based approaches have been proposed for single-station [5] and multi-station assembly systems (OSER-MAS) [6] that have integrated Bayesian Deep learning with CAE simulations, hence enabling effective root cause analysis (RCA). The OSER-MAS leverages a Bayesian 3D U-Net encoder-decoder based architecture with multiple output heads. Multiple output heads enable simultaneous estimation of a heterogeneous set of process parameters that can be real-valued or categorical while quantifying uncertainties. The decoder is leveraged to estimate object shape errors for upstream stations. The OSER-MAS gives superior performance for tasks such as process parameter estimation, object shape error estimation and uncertainty quantification but lack capabilities for scalability and interpretability.

This paper proposes a novel *closed-loop in-process (CLIP)* diagnostic framework which simultaneously enhances scalability and interpretability of the current OSER-MAS based approach by integrating an algorithm portfolio inclusive of techniques such as closed-loop training, transfer learning [7], continual learning [8] for scalability and 3D gradient-based class activation maps (3D Grad-CAMs) [9] for

interpretability within the OSER framework. The CLIP diagnostic framework shortens OSER-MAS model training time by developing: (i) closed-loop training to enable faster convergence for a single MAS by leveraging uncertainty estimates of the OSER-MAS model; and, (ii) scalability model to transfer meta-knowledge from the trained model to a new MAS and thus, each new MAS requires comparatively less training samples. Additionally, CLIP increases the OSER-MAS transparency for their quality root cause predictions by developing an interpretability model which entails: (a) linking elements of MAS model with functional elements of the Bayesian deep learning architecture; and, (b) relating features extracted by the Bayesian deep learning architecture with elements of the MAS model and further with the object shape error patterns for root causes(s) that occur in MAS. Although the CLIP framework is developed and validated considering the previously proposed OSER based approaches, the framework can be leveraged to simultaneously enhance the scalability and interpretability of similar approaches in related domains such as stamping and machining where deep learning and CAE simulations are leveraged to relate object shape errors to root causes.

## B. RELATED WORK

### 1) SCALABILITY

Scalability within manufacturing systems has been stated as a set of capabilities to provide transfer of knowledge and ideas from other engineering and management areas [10]. Scalability for algorithms to perform RCA of MASs translates into effectively leveraging the learning for one type of assembly system and then transferring this learning in form of features and relationships which can be relevant for another similar assembly system, and hence can enable learning for the latter using an exponentially lesser amount of data and computation capabilities. Applications of using transfer learning techniques have been done for fault diagnosis [11]. Digital Twins [12] have also been proposed as a way to enable scalability. Successful applications of transfer learning across multiple domains [7], [13]–[16] have enabled scalability in a sustainable manner that does not require exhaustive training data and computation capabilities. Recent works have also proposed that scalability should be life-long or continual and should not come at an expense of forgetting previous learning when new features or relationships are learnt for new systems [8], [17]–[20]. Hence, to enable scalability for MAS frameworks and methodologies that integrate transfer/continual learning with existing deep learning approaches for RCA, it is essential to ensure that training data and computation times do not become barriers for the application of such models within industrial setups.

### 2) INTERPRETABILITY

Interpretability has been another major concern for the application of deep learning-based RCA models within MAS as they do not provide the required context, trust and

confidence within root cause estimates. This lack of transparency when coupled with costly actions driven by them result that such models are not being adopted at scale. Various methodologies such as Gradient-based class activation maps that can integrate deep learning estimates with the required transparency have been proposed [21]–[23]. Bayesian deep learning [5] has been proposed to integrate confidence and uncertainty measures with root cause estimates but there is a need for frameworks within MAS that can provide interpretability while accounting for context and confidence. Such frameworks enable trust in black-box deep learning models and provide different levels of interpretability.

### 3) ROOT CAUSE ANALYSIS OF ASSEMBLY SYSTEMS

Dimensional and geometric variations are some of the biggest challenges faced by the manufacturing industry. Indeed, two-thirds of quality issues in the automotive and aerospace sectors are caused by dimensional variations [24]. Driven by the development of in-line measurement systems such as coordinate measuring machines (CMM) and 3D scanners, models for RCA of assembly systems have seen a lot of development in both, industrial applications, and academic research. These models [25] can be grouped into two categories: (a) knowledge-based models; and (b) estimation-based data-driven models leveraging statistical and machine learning techniques [26]. For single station systems, Apley and Shi [27] established the deviation transfer model based on process information such as fixture positioning and used least-squares to diagnose fault sources. Chang et al. [28] leveraged a linear model between shape error sources and measurement features followed by parameter estimation and statistical tests to diagnose shape error sources. Yu et al. [29] leveraged influence coefficients based on finite element modeling to establish shape errors between the sources of variation and measurements of flexible sheet metal parts. Further least-squares estimation was used to estimate errors in fixture positioning.

For MASs, Agrawal et al. [30] used regression models of sensor data. Zou et al. [31] proposed integrating BIC with LASSO variable selection. Shang et al. [32] proposed a Binary State Space Model (BSSM) for MASs to perform binary diagnosis. Jin et al. [33] proposed state-space and stream-of-variation (SoV) for multi-station shape error propagation of automotive assemblies. Ding et al. [34] extended the SoV method of assembly shape error of rigid parts using state space considering different fixture locating scheme. Using the above approaches as a base, various RCA models for MASs have been proposed. Ding et al. [35] [36] compared different variance estimation techniques and concluded a basis for method selection under different scenarios. Ceglarek and Prakash [37] proposed shape error diagnosis based on enhanced piecewise least squares (EPLS) to detect and isolate collinear dimensional faults caused by fixture variation. Ceglarek and Shi [38] [39] employed pattern matching for diagnosis of fixtures based on principal component analysis. Liu and Hu [40] used designated component analysis for shape

error diagnosis of flexible sheet metal parts. Various enhancements have been proposed using the knowledge of MASs [41] [42].

Given the ever-increasing complexity of MASs, increased computation capabilities and developments in machine learning, recently, RCA approaches [26] using machine learning have been proposed to overcome limitations of the above-stated methods such as linear approximations of the MAS. Du et al. [43] utilized artificial neural networks to monitor and identify process variability. Beruvides et al. [44] applied reinforcement learning to perform RCA. Bayesian Networks [45], [46] are seen as an alternative to solve small dataset problems and integrate process data and engineering knowledge. Recently, Sinha et al. developed Object Shape Error Response (OSER) for single-station [47][5] and Object Shape Error Response for Multi-Station Assembly Systems (OSER-MAS) [6] that aim to integrate Bayesian deep learning elements such as Bayesian 3D Convolutional Neural Networks and Computer-Aided Engineering (CAE) simulations thereby, blending (a) engineering knowledge-techniques with (b) estimation-based data-driven approaches. This satisfies various model capability requirements for RCA of MASs such as (i) high data dimensionality [48]; (ii) non-linearity [49]; (iii) collinearities [50]; (iv) high faults multiplicity [51]; (v) uncertainty quantification [52]; (vi) dual data generation capabilities [12]; (vii) high dimensionality and heterogeneity of process parameters [53]; and, (viii) fault localization [54]. In addition to the aforementioned model capability requirements, RCA techniques must be further developed and enhanced to fulfil two additional key requirements [55] in order to enable implementation and large-scale adoption across different manufacturing environments:

(ix) *Scalability* as automotive multi-station assembly processes include hundreds of stamping parts and components, multiple stations with multiple stages in each station [4] namely, place-clamp-fasten-release (PCFR) to finish the final assembly product. The multiple variation sources in the MAS interact and accumulate in a non-additive manner. The final product accuracy and performance depend upon the accumulated performance of individual stations in the system. In MASs, the quality of the components is influenced by (i) incoming non-ideal and deformable parts; (ii) PCFR-to-part interactions as parts move through PCFR stages (shape error induced by fixturing and joining operations); (iii) part-to-part interactions within each station and further magnified between stations; and, (iv) station-to-station interactions due to re-orientation errors between stations (change of fixture locating layout between stations).

(x) *Interpretability* as the estimations of root cause(s) will require insights into why such estimates were made by the deep learning model. Such interpretability insights are essential for contextualizing the root cause(s) estimated by the deep learning model. Additionally, root cause (RC) estimates drive costly corrective actions hence, model interpretability

integrated with measures of uncertainty [5] are crucial requirements for effective and efficient corrective and control actions.

The paper will address the aforementioned requirements as follows:

*Requirement (ix)* by developing a closed-loop training framework that leverages the epistemic uncertainty [5] estimates of the Bayesian 3D U-Net based OSER-MAS model to intelligently sample from the process parameter hyperspace for faster convergence and hence, reduce the computation time bottleneck of the CAE simulations. The presented CLIP diagnostic framework in this paper will utilize high fidelity CAE simulator of the assembly process called Variation Response Method (VRM) [56]. Further, to exponentially enhance the scalability for high dimensional MAS and reduce CAE simulation time, uncertainty guided continual learning [8] and transfer learning [7] features are integrated with the CLIP diagnostic framework underpinned algorithm portfolio. This enables the transfer of meta-knowledge from the trained model to a new MAS and thus, each new MAS requires comparatively less training samples. Theoretically, given that multi-physics processes of the assembly system are similar within each station hence, the features extracted by spatial convolutional operations are transferable across different assemblies; thus, making transferability within the model essential for scalability. Given that models for different MAS would be trained sequentially, leveraging continual approaches reduces catastrophic forgetting. A model with continual learning capabilities can also account for the dynamic nature of the assembly system and hence, achieve life-long learning. This is accomplished using uncertainty guided continual learning [8] that leverages the Bayesian neural network parameter uncertainty to assign importance for each task i.e., a particular assembly case study thereby, enabling continual learning by updating less important parameters at a faster rate.

*Requirement (x)* by developing an interpretability model which is based on 3D Gradient-weighted Class Activation Maps (3D Grad-CAMs) and entails: (a) linking elements of MAS model with functional elements of the 3D Bayesian U-Net model; and, (b) leveraging 3D Grad-CAMs [9] that provide insights into the regions within the input that the model focuses on to estimate process parameters i.e. root cause(s). These collectively provide the required

interpretability for RCA. Additionally, Bayesian Deep Learning enables uncertainty quantification and segregation into aleatoric and epistemic uncertainties, thus, providing a measure of the required confidence while conducting RCA.

The key contribution of the paper is the development of the CLIP diagnostic framework underpinned algorithm portfolio that includes the following models:

- (1) A closed-loop training model to enable faster convergence for a single MAS by leveraging uncertainty estimates of the Bayesian 3D U-net OSER-MAS model.
- (2) Uncertainty guided transfer/continual learning-based scalability model to transfer meta-knowledge from the trained model to a new MAS and thus, each new MAS requires comparatively less training samples.
- (3) A 3D Grad-CAMs based interpretability model which links functional elements and features extracted by the 3D U-Net architecture with elements of the MAS model and further with the shape error patterns for root causes(s) that occur in MAS.
- (4) Verify and validate the scalability and interpretability capabilities of the CLIP diagnostic framework underpinned algorithm portfolio using six different industrial automotive assemblies of varying complexities.

The rest of the paper is organized as follows; Section II formulates the object shape error estimation and RCA problem for MASs, Sections III and IV discuss the methods for scalability and interpretability, respectively, Section V presents the industrial case studies and finally, conclusions and future work are summarized in Section VI.

## II. PROBLEM FORMULATION

### A. MULTI-STATION ASSEMBLY SYSTEMS

As illustrated in Fig. 1 [6], MAS can be represented as a process tree where different nodes correspond to stages within a single assembly station (Fig. 1a) or as stations within the assembly system (Fig. 1b). A station consists of multiple stages namely, positioning, clamping, fastening and release (PCFR). The input to each station is a set of incoming parts (objects) that need to be assembled. Within the process, object shape errors can be induced in any station by one or multiple variations of the process parameter(s). These errors are further propagated and accumulate in a non-additive manner [34]. Any variation in the process parameter(s) is a source of shape

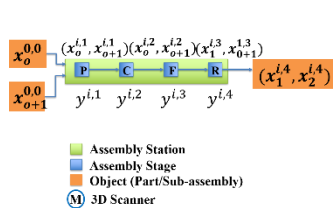


FIGURE 1a. PCFR stages of an assembly station

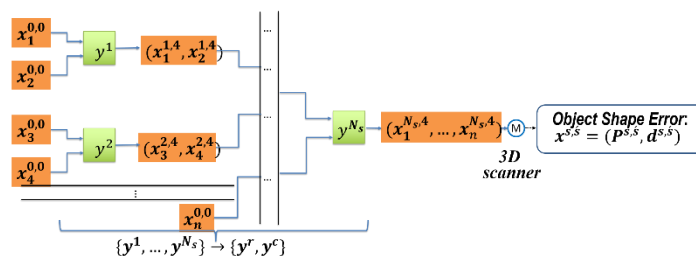


Figure 1b. Multi-station assembly system with  $N_s$  stations

Figure 1. Multi-station assembly system with PCFR assembly stages



error and thus must be first quantified and then estimated as a root cause(s) of the shape errors. In MAS, these process parameters are classified into three categories: (a) Real-valued parameters of incoming parts (objects) variation as caused by upstream fabrication processes such as stamping, extrusions, etc.; (b) Real-valued process parameters related to PCFR stages of each assembly station. They represent any deviation from nominal in fixturing/tooling or joining operations; and, (c) Binary joining-based process parameters in the fastening stage that indicate the success of the joint. The value is  $\{1\}$  when joint is successfully completed or  $\{0\}$  for an unsuccessful joint due to the excessive gap between objects to be joined or current failure in the tool. In this paper, Self-Piercing Riveting (SPR) is the considered fastening /joining process.

Stations in the MAS are represented by  $s: s = 1, \dots, N_s$ , where  $N_s$  represents the total number of stations within the system and  $\acute{s}: \acute{s} = 1, 2, 3, 4$  represents the four stages within each station. Any object  $o: o = 1, \dots, n$  at its design nominal shape is characterized by a set of nominal points  $\mathbf{P}_o = \{\mathbf{p}_{ok}\}$ ,  $k = 1, \dots, n_o$ , where  $\mathbf{p}_{ok}$  is a vector consisting of the x, y and z coordinates of the kth input point and  $n_o$  represents the total number of points on object  $o$ .  $\mathbf{d}_o = \{\mathbf{d}_{ok}\}$  denotes the deviation of each point  $k$  after the nominal object  $o$  has gone through different stages of the process,  $\mathbf{d}_{ok}$  is a vector comprised of deviations of each point in x, y and z axes on object  $o$ . As the object  $o$  goes through multiple stations and stages within the stations, the set of points are represented as  $\mathbf{P}_o^{s,\acute{s}}$  while  $\mathbf{d}_o^{s,\acute{s}}$  represents the deviations. Object shape error for object  $o$  after station  $s$  and stage  $\acute{s}$  can be represented as:

$$\mathbf{x}_o^{s,\acute{s}} = (\mathbf{P}_o^{s,\acute{s}}, \mathbf{d}_o^{s,\acute{s}}) \quad (1)$$

where  $\mathbf{x}_o^{0,0} = (\mathbf{P}_o^{0,0}, \mathbf{d}_o^{0,0})$  represents the incoming non-ideal part inclusive of the part variations and  $(\mathbf{x}_1^{s,\acute{s}}, \dots, \mathbf{x}_n^{s,\acute{s}})$  represents a set of objects (sub-assembly) after stage  $\acute{s}$  of station  $s$ .

On the other hand, the aforementioned process parameters which can be real-valued or binary are represented by  $\mathbf{y}^{s,\acute{s}}$  for station  $s$  stage  $\acute{s}$ . The process parameters for station  $s$ , inclusive of all stages, is represented by  $\mathbf{y}^s$  while  $\mathbf{y}$  represents the total set  $h$  of process parameters across the entire assembly system. The process parameter vector  $\mathbf{y}$  consists of  $c$  binary parameters and  $r$  real-valued parameters denoted by  $\mathbf{y}^c$  and  $\mathbf{y}^r$ , respectively.

The aim of the proposed 3D U-Net CNN model training is to learn a function  $f(\cdot)$  that takes as input the combined object shape error at the end of the system  $\mathbf{x}^{N_s,4}$ , i.e., after the final stage of the last station ( $N_s$ ), and estimates the process parameters across the entire system and the object shape error for all objects at the end of the previous stations:

$$[\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^{1,4}, \dots, \mathbf{x}^{N_s-1,4}] = f(\mathbf{x}^{N_s,4}) \quad (2)$$

## B. ROOT CAUSE ANALYSIS

For comprehensive RCA for MASs, the paper proposes three key steps namely: (i) fault identification; (ii) fault localization; and, (iii) fault isolation. Using the estimates obtained within (2), RCA can be done using the following steps to isolate single or multiple faults that occur in a MASs.

(i) *Fault Identification*: Fault identification involves identifying which process parameters are potentially at fault. Faults can be identified by comparing the values of process parameters with given standards. For all binary process parameters  $\mathbf{y}^c$  that are  $\{0\}$  are identified as faults as they represent a failure in the joining process. Real-valued process parameters can be identified as potential faults based on the fault identification strategy. If a threshold approach is leveraged, each  $\mathbf{y}^{s,\acute{s}}$  that is greater than the threshold is identified as a fault on the other hand if six-sigma fault identification strategies are used a sample of products  $\mathbf{s}_p$  are observed and mean  $\mu_p^{s,\acute{s}}$  and standard deviation  $\sigma_p^{s,\acute{s}}$  of each process parameter is calculated. Based on the significance level used, a mean shift or a change in variance (heteroskedasticity) in process parameters can be identified as a fault. The subset of process parameters identified as fault via a threshold or a six-sigma approach can be denoted as:

$$\mathbf{y}_F \subseteq \mathbf{y} \quad (3)$$

(ii) *Fault Localization*: Fault localization involves identifying particular stations within which the object(s) (sub-assemblies) shape error becomes significant. The shape error estimates  $[\mathbf{x}_o^{1,4}, \dots, \mathbf{x}_o^{N_s-1,4}]$  for all objects  $o: o = 1, \dots, n$ , are compared with the design nominal. If the shape error is beyond the threshold (assembly tolerances) at the end of a station and within the threshold, for the previous station the fault is localized to that particular station for the corresponding object. This is done for each object and thus, multiple faults for different objects can also be localized. The subset of stations localized is denoted as:

$$\mathbf{s}_F \subseteq \mathbf{s}, \mathbf{o}_F \subseteq \mathbf{o} \quad (4)$$

(iii) *Fault Isolation*: Fault isolation involves integrating the information from fault identification and fault localization to isolate which process parameters within the potentially identified faults  $\mathbf{y}_F$  and localization stations  $\mathbf{s}_F$  are ‘malignant’ and have a significant impact on the shape error of the final object (product). For all potentially identified process parameters, the process parameters that lie within the localized stations  $\mathbf{s}_F$  (and corresponding objects  $\mathbf{o}_F$ ) are isolated as faults and estimated as RCs. Given that manufacturing systems are stochastic, all process parameters always have an inherent level of variation, hence, using such a three-step approach enables differentiating faults that are benign (have no significant impact on the product shape error) from faults that are malignant (cause product shape error to go beyond assembly thresholds). The RCs can be denoted as:

$$RC \subseteq \mathbf{y}_F \forall \mathbf{y}_F \subseteq \mathbf{S}_F \quad (5)$$

### III. METHODS FOR SCALABILITY

#### A. BAYESIAN 3D U-NET ARCHITECTURE

For the estimation of  $f(\cdot)$  as shown in (2), Sinha et al. [6] proposed the Object Shape Error Response for Multi-Station Assembly Systems (OSER-MAS) approach. The proposed approach leverages a Bayesian 3D U-Net architecture (Fig. 2) [6] that enables: (i) estimation of a heterogeneous set of process parameters; (ii) estimation of upstream object shape errors; and, (iii) quantification and segregation of uncertainties. The model is trained using a weighted loss function [6] that accounts for all the aforementioned outputs and uncertainty quantification. The architecture consists of four levels of the encoder-decoder. The end of the encoder consists of a regression and classification head, each contains one hidden Dense Flipout layer with 64 nodes and ReLU activation. The output nodes in each head are equal to the real-valued and binary process parameters, respectively. These heads enable estimation of a heterogeneous set of process parameters. The end of the decoder estimates the upstream object shape errors. Given the use of Bayesian Flipout layers in the encoder and regression and classification heads, the architecture enables quantification and segregation of uncertainties. The encoder-decoder model consists of seven key functional elements namely: (1a) Object shape error voxelization; (1b) Encoder with down-sampling kernels; (1c) Decoder with up-sampling kernels; (1d) Multiple output heads; (1e) Attention gate; (1f) Bayesian Flipout layers; and, (1g) Residual connections. The interpretability of each element is discussed concerning the requirements of MASs. Overall the model takes as input the voxelized shape error after the final station  $\mathbf{x}^{N_s,4} \rightarrow \{\mathbf{V}_{u,v,w,d}\}$  and give as output the shape error after previous stations and the process parameters

$[\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^{1,4}, \dots, \mathbf{x}^{N_s-1,4}]$ . Uncertainties are estimated for each output value. The uncertainty estimates are crucial in driving costly corrective actions. They are segregated into epistemic and aleatoric uncertainties. The aleatoric uncertainty is estimated by considering the outputs follow a multivariate normal distribution with a diagonal covariance matrix. The epistemic uncertainty is estimated by assuming each weight  $\omega$  in the network follows a normal distribution with parameters  $\theta_\omega = (\mu_\omega, \sigma_\omega)$ , and then estimating the posterior parameters of the distributions. The paper proposes to leverage these measures of uncertainty to build approaches that enable scalable learning. The overall epistemic uncertainty of the estimated process parameters  $\sigma(\mathbf{y})$  and the uncertainty of the weights  $\sigma_\omega$  can be further leveraged to build methods that enable scalable learning by leveraging closed-loop sampling from CAE simulators and further leverage uncertainty-guided continual learning [8] for effective learning that aids in transferring knowledge in between similar MASs hence enabling convergence using exponentially lesser training samples while also ensuring that there is no catastrophic forgetting of learning for previous MASs.

#### B. CLOSED-LOOP SAMPLING AND TRAINING

Closed-loop sampling enables the dynamic and adaptive generation of training samples based on the uncertainty and error of the previous training iterations while ensuring that the sample generation has a degree of randomness to prevent the repeated generation of similar samples (Table I). This enables faster convergence to the optimal weights and biases distribution parameters of  $f(\cdot)$  as shown in (2). Sampling is done using VRM [56] as the CAE simulator, which takes as input a set of process parameters and outputs the object shape errors after the desired stage/station. Initially, Latin Hypercube Sampling (LHS) [57] of the process parameters

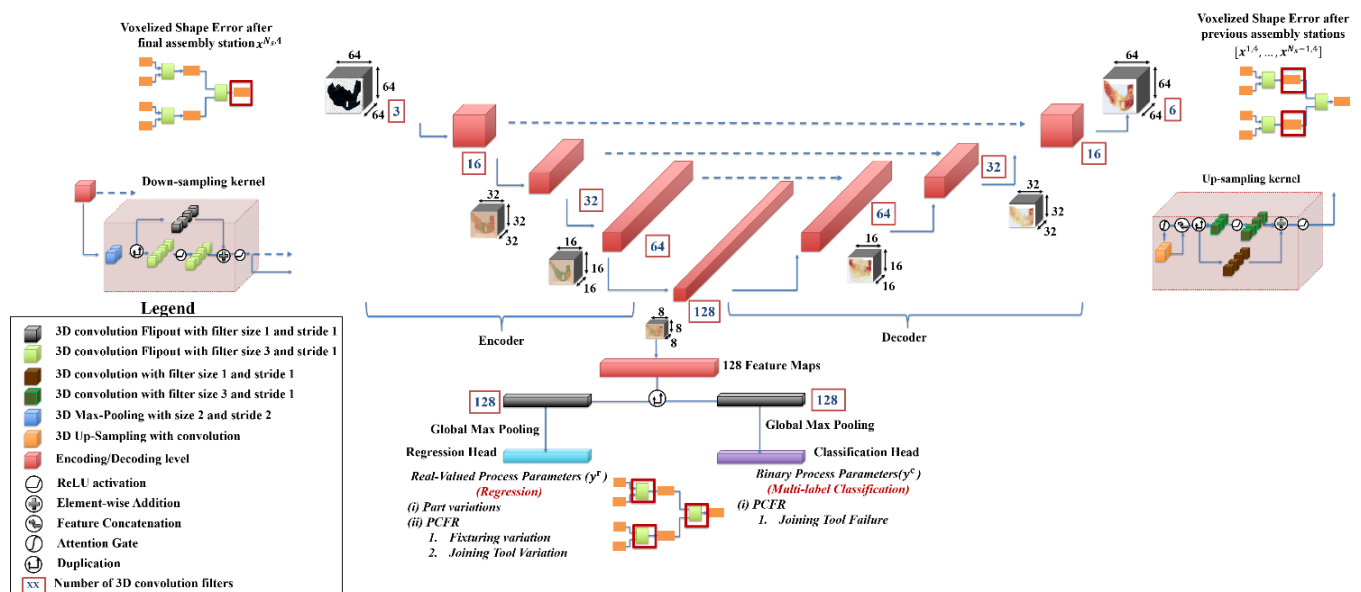


FIGURE 2. U-Net Based Encoder Decoder Architecture, left: down-sampling kernel, right: up-sampling kernel

within the allowable ranges is done for input to the CAE simulation model to generate the test (**E**) and validation (**V**) set.

$$\mathbf{E} = (\mathbf{X}^E, \mathbf{Y}^E), \mathbf{V} = (\mathbf{X}^V, \mathbf{Y}^V) \quad (6)$$

While each element in  $\mathbf{X}$  is characterized by the object shape error  $\mathbf{x}^{s,s}$  (1) and each row in  $\mathbf{Y}$  consists of a vector of process parameters:  $\mathbf{y} = \{\mathbf{y}^r, \mathbf{y}^c\}$ . The initial training set  $\mathbf{T}^0$  is also generated using LHS. This is used to train the proposed architecture  $f(\cdot)$ . After training inference is performed on the validation set to obtain the predictions and uncertainty on the validation set (**V**)

$$[\hat{\mathbf{Y}}^V, \sigma^V] = f(\mathbf{X}^V) \quad (7)$$

For each sample the absolute error is calculated and summed up across all the process parameters:

$$\mathbf{E}^v = |\hat{\mathbf{Y}}^V - \mathbf{Y}^V| \quad (8)$$

The error is summed up across all  $h$  process parameters for each sample  $\mathbf{e}^v$  which is a column vector consisting of combined error for each sample.

$$\mathbf{e}^v = \sum_h \mathbf{E}^v \quad (9)$$

The normalized error ( $\tilde{\mathbf{e}}^v$ ) and uncertainties ( $\tilde{\sigma}^v$ ) are weighted to obtain the sampling importance metric  $\tau$  for each sample:

$$\tau = w \cdot \tilde{\mathbf{e}}^v + (1 - w) \cdot \tilde{\sigma}^v \quad (10)$$

Samples are sorted based on the sample metric. This is done considering that the samples having the highest importance, i.e., having the maximum sum of error and uncertainty would have a more significant contribution to model convergence than other samples. Based on sampling metrics and actual process parameters  $\mathbf{Y}^V$ , the parameters of the sampling distribution are estimated. Gaussian Mixture Model (GMM) with a pre-specified number of mixtures is considered as the sampling distribution. The sorted set of sampling metric and process parameter is represented as  $(\tilde{\mathbf{r}}^v, \tilde{\mathbf{Y}}^v)$ . Given the GMM has  $K$  mixtures, the sorted set is subdivided into  $K$  blocks each block  $i^b$  having samples  $\tau_{i^b} = \dim(V)/K$  where  $\dim(V)$  represents the total number of samples within the validation set. Each block  $i^b$  is used to estimate distribution parameters for the  $i^b$ th mixture. Each mixture is characterized by a multivariate normal distribution while the component within the multi-variate normal corresponds to the process parameter. The distribution parameters for the  $i^{bth}$  component consisting of  $s$  samples is estimated as in (11), where  $\mathbf{Y}_{i^b}^V$  represents the subset of samples in the block  $i^b$

$$\phi_{i^b} = \sum_s \tilde{\mathbf{r}}^v, \mu_{i^b} = \text{Mean}_s(\mathbf{Y}_{i^b}^V), \Sigma_{i^b} = \text{Covar}_s(\mathbf{Y}_{i^b}^V) \quad (11)$$

Where  $\phi$  vector is normalized, to sum up to 1 and hence represent mixture component weights  $\phi_{i^b}$ , and where  $\mu_{i^b}, \Sigma_{i^b}$  represent the mean vector and covariance matrix respectively for the  $i^{bth}$  mixture. After estimating the

distribution,  $\mathbf{T}^i$  samples are drawn and evaluated and then added to the training set to further train the model. This ensures that samples are drawn considering the error and uncertainty while accounting for the fact that the model should not over fit on the validation set. It should be noted that although the validation set is used for sampling, the model is never trained on the validation samples, the randomness in the sampling ensures that samples are drawn from regions where the error and uncertainty are high. After each training is done the model is tested on the test set (**E**) to determine the model performance. The training and sampling are terminated either after i) the performance on the test set (**E**) reaches the required threshold  $\varepsilon$  or ii) the maximum number of training iterations  $n^m$  is reached. The performance threshold can be decided based on the case study and application, and the maximum number of iterations is decided based on the CAE simulation budget.

TABLE I  
CLOSED LOOP SAMPLING & TRAINING FRAMEWORK

**Algorithm 1:** Closed-loop sampling and training

<b>Procedure:</b>
Generate and evaluate Testing ( <b>E</b> ) and Validation ( <b>V</b> ) set
Generate Initial Training Set ( $\mathbf{T}^0$ )
<b>Repeat</b> $i = 1$ <b>to</b> $n^m$ :
Train model on training set ( $\mathbf{T}^i$ )
Perform inference on validation ( <b>V</b> ) set
Estimate GMM parameters ( $\phi, \mu, \Sigma$ ) using (8), (9) and (10)
Sample from the GMM and evaluate $\mathbf{T}^s$
Add samples to training set $\mathbf{T}^{i+1} \leftarrow \mathbf{T}^i \cup \mathbf{T}^s$
<b>Stopping Checks:</b>
Error on test ( <b>E</b> ) set, below the threshold $e^e < \varepsilon$
Maximum number of iterations reached $i = n^m$

### C. UNCERTAINTY GUIDED CONTINUAL LEARNING

The closed-loop sampling approach enables faster convergence within one MAS but does not enable the trained  $f(\cdot)$  to be used across different MASs. Leveraging the trained function  $f(\cdot)$  across different MASs to enable transferring of relevant knowledge (features) and hence enable convergence in comparatively lesser samples is crucial in enabling scalability. Continual learning methods (also known as sequential/lifelong learning) aim to incrementally learn new tasks without forgetting previous tasks for which they have been trained. In the context of MASs, the scale-up starts from tasks or cases as simple as a coupon or Top-hat assembly to full-scale MASs such as automotive car door or cross member assemblies and can cumulatively consist of up to 100 assembly stations each consisting of multiple stages. Each assembly case is treated as a task  $T_i$ , and continual learning is performed for a total of  $T_n$  tasks. Continual learning enables the transfer of the process parameter estimation capabilities of previous assembly cases to more complex assemblies while retaining the essential capabilities required for process parameter estimation of previous assemblies. The key to achieving continual learning requires assignment of importance to each neural network weight  $\omega$  and further updating only non-

important weights such that the model learns the new task without forgetting the previous task [8]. The approach leverages uncertainty guided continual learning because the weight uncertainty  $\sigma_\omega$  of the Bayesian 3D U-Net model serves as an implicit measure of importance. Additionally, the ease of interpretation, strong mathematical foundation and good results on various datasets [8] motivate the use of such a learning algorithm.

Given the use of Bayesian neural networks and a normal distribution parametrized by  $\theta_\omega = (\mu_\omega, \sigma_\omega)$  for each weight  $\omega$  within the network, the standard deviation of each weight distribution is leveraged as the metric for importance. To enable continual learning the learning rate  $\alpha$  for each parameter is updated by the corresponding importance  $\Omega$ .

$$\alpha_\mu \leftarrow \alpha_\mu / \Omega_\mu, \alpha_\sigma \leftarrow \alpha_\sigma / \Omega_\sigma \quad (12)$$

The importance of the parameters is set to be inversely proportional to the standard deviation, which mathematically translates that weights with higher standard deviation are less important and hence can be updated at a higher rate to learn new tasks, while weights with lower standard deviation are more important and hence should be updated at a lower rate to prevent catastrophic forgetting (performance loss) for the old tasks.

$$\Omega \propto 1/\sigma_\omega \quad (13)$$

Based on various empirical studies done by Ebrahimi et al. [8], the learning rate adaptations were determined as:

$$\Omega_\mu \leftarrow 1/\sigma_\omega, \Omega_\sigma \leftarrow 1 \quad (14)$$

The overall algorithm consisting of closed-loop sampling and continual learning, to train the model on multiple tasks  $T_1, \dots, T_n$  (different assembly case studies) is shown in Table II. After each task, the learning rates are updated as shown in (14). The number of output nodes within the model is kept equal to sum all process parameters across all cases studies. For each assembly case, the specific process parameters nodes output the values while other nodes corresponding to process

TABLE II  
CONTINUAL LEARNING FRAMEWORK

**Algorithm 2:** Uncertainty based continual learning with closed-loop sampling

<b>Procedure:</b>
Setup all tasks (Assembly Case Studies)
Set initial learning rate $\alpha_\mu = \alpha_\sigma = \alpha_0$
<b>Repeat for all tasks <math>T_n</math>:</b>
Perform closed loop training for $T_i^{\text{th}}$ task (Algorithm 1)
<b>Update Learning Rates:</b>
$\Omega_\mu \leftarrow 1/\sigma_\omega$
$\Omega_\sigma \leftarrow 1$
$\alpha_\mu \leftarrow \alpha_\mu / \Omega_\mu$
$\alpha_\sigma \leftarrow \alpha_\sigma / \Omega_\sigma$
<b>end</b>

parameters for other assembly cases are set to output a nominal fixed value (generally set to zero). Overall continual learning aims to learn process parameter estimation capabilities of each assembly case study incrementally while minimizing the forgetting for previous assembly case studies.

#### D. TRANSFER LEARNING

Transfer learning is an effective method for transferring learning (process parameter estimation for MASs) from one task to related tasks (between different assembly case studies) using exponentially lesser training samples and hence acts as a key enabler for scalability. The paper leverages transfer learning and continual learning as a combined algorithm portfolio enabling scalability. The choice between them can be made based on training results and deployment performance.

Transfer learning is mathematically formalized [14][58] as a domain  $D$  which consists of features  $X$  and a distribution over the feature space  $P(X)$ . In this case the domain entails process parameter estimation  $f(\cdot)$  on a particular assembly case  $T_i$  with shape error features  $X$  and distribution over the feature space as  $P(X)$ . Within Domain  $D$  task  $T$  is performed that constitutes learning a conditional distribution  $P(Y|X)$  to estimate process parameters  $Y$

$$D = \{X, P(X)\} \quad (15)$$

The paper aims to ‘transfer learn’ from the source domain ( $D_s$ ) corresponding to a particular assembly case to a target domain ( $D_T$ ), i.e. a similar assembly case to perform the same task of estimating  $f(\cdot)$  (2) while accounting for differences between cases such that at least one of the elements between the domain and target are not the same:

$$(X_s \neq X_T), (P(X_s) \neq P(X_T)), (Y_s \neq Y_T), P(Y_s|X_s) \neq P(Y_T|X_T) \quad (16)$$

Considering the prior knowledge on the similarity of assembly cases studies it can be estimated that: (i)  $(X_s \approx X_T)$  – given that similar features [59] need to be extracted from the object shape error data that include bends, twists, rotations, translations etc.; (ii)  $(P(X_s) \approx P(X_T))$  – similarly, the distribution around the input features is approximately the same; (iii)  $Y_s \neq Y_T$  – the outputs for each study are different given a different number of process parameters are involved; and, (iv)  $(P(Y_s|X_s) \neq P(Y_T|X_T))$  – the conditional distribution is also significantly changed given the change in the assembly system and the output. To account for (iii) the final layer of the network is replaced with nodes corresponding to the new set of process parameters; to account for (iv) the approach leverages standard protocols established in transfer learning to achieve transferability.

Based on past work on successful applications of transfer learning that involved using ImageNet data to aid Computer-Aided Detection [15], the fine-tuning transfer learning protocol is leveraged. The network weights are initialized



using the weights of a network trained on the previous assembly case study. The whole network is then fine-tuned while keeping the learning rate of the convolutional layers  $\alpha_c$  in the first two encoder levels ten times less than the rest of the network  $\alpha_f$ . The regression and classification heads are replaced and reinitialized. The nodes in each of the heads are determined by the number of process parameters for the case study. Overall transfer learning (Table III) aims to learn process parameter estimation capabilities of each assembly case study while ‘transferring’ knowledge from previous case studies. Although while doing this the model can ‘forget’ estimation capabilities for previous case studies. Fig. 3 summarizes the overall framework for closed-loop sampling and training integrated with continual and transfer learning approaches.

TABLE III  
TRANSFER LEARNING FRAMEWORK

*Algorithm 3: Transfer learning with closed-loop sampling*

<b>Procedure:</b>
Setup all tasks (Assembly Case Studies)
Set initial learning rate $\alpha_0$ for all layers
Perform closed-loop training on 1 <sup>st</sup> task
Set learning rate lower convolutional layers $\alpha_c \leftarrow \alpha_0/10$
Set learning rate for all other layers $\alpha_f \leftarrow \alpha_0$
<b>Repeat for all tasks (2 to <math>T_n</math>):</b>
Replace output layer of neurons equal to the number of process parameters for the $T_i^{\text{th}}$ task
Perform closed loop training for $T_i^{\text{th}}$ task (Algorithm 1)
Initialize network with trained weights
<b>end</b>

## IV. METHODS FOR INTERPRETABILITY

### A. 3D U-NET ARCHITECTURE INTERPRETABILITY

The implementation of deep learning models within industrial environments requires opening the black box and providing interpretability and causation on why the deep learning models can give superior performance as compared to traditional linear or piece-wise linear approaches traditionally used for RCA of MASs. The paper proposes to do that on two levels:

Providing a link between MASs requirements and functional elements of the Bayesian 3D U-Net architecture of the OSER-MAS approach. This also aims to provide a link between the engineering challenges faced in RCA of MASs and the developments done within the OSER-MAS model to overcome these challenges.

Leveraging 3D Grad-CAMs to interpret the features that are extracted by the architecture and then propagated through various encoder and decoder layers to be interpreted as root causes. To integrate high measures of confidence within the deep learning model estimates, it must be established that the input context  $\mathbf{x}^{N_s,4}$  (object shape error) on which the model focuses should be directly related to the estimated output  $\mathbf{y}$  (process parameter or root cause), e.g. if the model estimates

‘part variation’ as a root cause, it should focus on the ‘part variation’ rather than other possible root causes such as ‘clamping’ or ‘positioning’. Clearly extracted semantics in convolutional layers integrates a much-needed measure of ‘trust’ within the root cause estimates.

### B. 3D GRADIENT-WEIGHTED CLASS ACTIVATION MAP

3D Gradient-weighted class activation maps (3D Grad-CAMs) aim to visualize the input features that led to a particular output. In the context of MASs, this aims to localize key regions within the input shape error  $\mathbf{x}^{N_s,4}$  that led to the estimation of a process parameter  $\mathbf{y}$ . This is estimated by taking a discriminative gradient of a particular process parameter  $\mathbf{y}^m$  output with respect to the feature map of a selected convolutional layer within the 3D U-Net architecture. The map for a particular output process parameter  $\mathbf{y}^m$  is represented as  $\mathbf{L}_{grad-CAM}^{y^m}$  and can be calculated as a weighted sum of the features maps:

$$\mathbf{L}_{grad-CAM}^{y^m} = \text{ReLU} \sum_{f=1}^F \alpha_f^{y^m} \mathbf{A}^f \in \mathbb{R}^{u \times v \times w} \quad (17)$$

Where  $\mathbf{A}^f$  represents  $f$  feature maps ( $f = 1, 2, \dots, F$ ) for the selected convolutional layer and  $a, b, c$  represent the dimensions of the 3D feature maps. ReLU represents the activation function which is rectified linear unit. The weights are calculated by summing the gradients for each element within the feature map:

$$\alpha_f^{y^m} = \frac{1}{abc} \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^c \frac{d\mathbf{y}^m}{d\mathbf{A}_{i,j,k}^f} \quad (18)$$

The  $\mathbf{L}_{grad-CAM}^{y^m}$  is interpolated to match the dimensions of the input voxelized object shape error. The overlay between the voxelized object shape error and interpolated 3D Grad-CAM provides interpretability information on what features/spatial regions within the shape error did the model focus on to estimate the selected process parameter  $\mathbf{y}^m$  of interest. The interpolated 3D Grad-CAM is then transformed to a point-based shape error and smoothing is done using a median filter for consistency across the mesh. These can be visualized to obtain regions within the shape error input that the neural network model is focusing on to estimate the process parameter.

Fig. 3 summarizes the algorithm portfolio including the integration of closed-loop training with continual/transfer learning-based scalability model and 3D Grad-CAMs based interpretability model. The effective integration of these models enhances the diagnostic capabilities of the OSER-MAS model and enables scalable and interpretable RCA.

## V. CASE STUDIES

### A. EXPERIMENTAL SETUP

Verification and validation is done using  $T_n = 6$  tasks or assembly systems (Fig. 4, Table IV) with varying

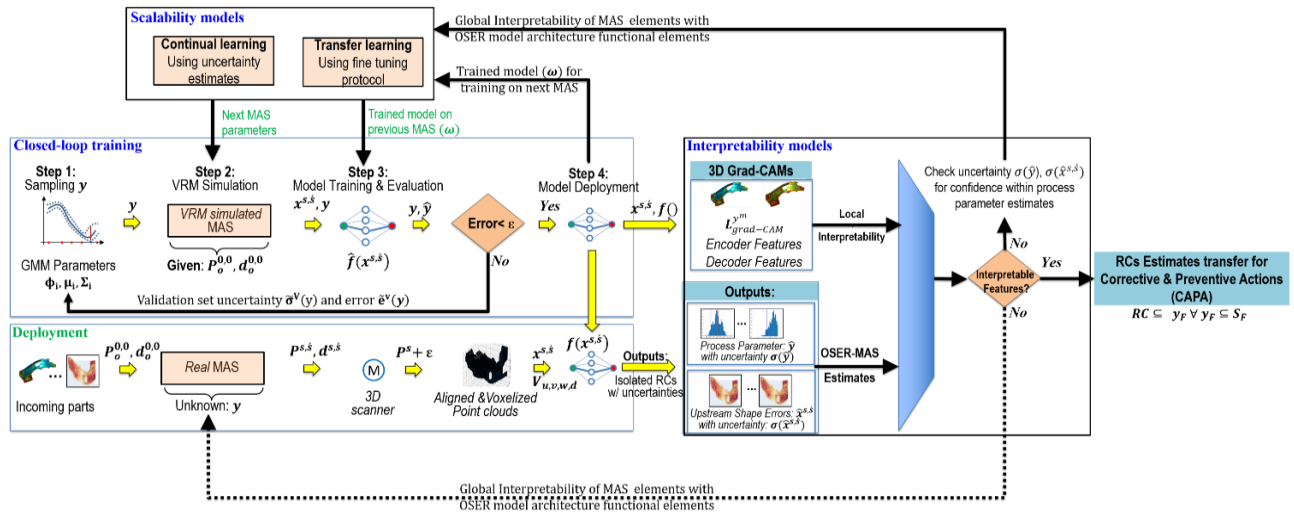


FIGURE 3. Closed-loop in-process (CLIP) diagnostic framework underpinned algorithm portfolio

complexities ranging from a single part coupon level assembly to automotive industrial multi-station assemblies. Each assembly case is considered as a unique task. Continual or transfer learning is done sequentially for all case studies as in the order mentioned below. The case studies include:

(1) *Flat Plate (Coupon) Assembly*: consists of  $n = 1$  ideal compliant part with a flat 2D geometry. It involves  $N_s = 1$  station and four stages (PCFR) and is controlled by  $h = 7$  real-valued  $y^r$  fixturing and joining based process parameters.

(2) *Top-Hat Assembly*: consists of a  $n = 2$  ideal compliant parts with a simple 3D geometry. It involves  $N_s = 1$  station and 4 stages (PCFR) and is controlled by  $h = 17$  real-valued  $y^r$  fixturing and joining based process parameters.

(3) *Door Halo Reinforcement Panel Assembly*: consists of  $n = 1$  ideal compliant part with complex 3D geometry. It involves  $N_s = 1$  station and 2 stages (PC) and is controlled by  $h = 3$  real-valued  $y^r$  fixturing based process parameters.

(4) *Door Inner and Hinge Reinforcement Assembly*: consists of  $n = 2$  ideal compliant parts with a complex 3D geometry. It involves  $N_s = 1$  station and four stages (PCFR) and is controlled by  $h = 6$  real-valued  $y^r$  fixturing based process parameters.

(5) and (6) *Cross Member Assembly*: consists of  $n = 4$  non-ideal compliant parts with complex 3D geometry. It involves  $N_s = 3$  stations each with four stages (PCFR). Two sub-cases within this are considered: case (5) consisting of  $h = 12$  real-valued  $y^r$  part variation and fixturing based process parameters; and, case (6) consisting of a heterogeneous (real-valued and binary) set of  $h = 158$  process parameters including 123 real-valued  $y^r$  part variation, fixturing and joining based process parameters and 25 binary process parameters  $y^c$  indicating the success of joining (Fig. 5).

For comparison and benchmarking of scalability, all five cases are analyzed under  $T_s = 4$  training scenarios are

considered:

(i) *Random Sampling*: Involves randomly sampling from the CAE simulator within the allowable ranges for each process-parameter. Each case study is trained on a re-initialized network with random weights. This also serves the baseline performance expectations.

(ii) *Closed-loop Sampling*: Involves training the five case studies using Algorithm 1 as shown in Table I.

(iii) *Transfer Learning with Closed-loop sampling*: Involves training the five case studies sequentially using Algorithm 3 as shown in Table III.

(iv) *Continual Learning with Closed-loop Sampling*: Involves training the five case studies sequentially using Algorithm 2 as shown in Table II.

Interpretability is verified by considering the cross member assembly (5) to provide links between MASs requirements and architecture functional elements and to obtain 3D Grad-CAMs for key process parameter variations. While obtaining 3D Grad-CAMs the weights and biases of the network are fixed at the mean values ( $\omega = \mu_\omega$ ).

Before training all shape errors are pre-processed and voxelized to  $(u, v, w, d) = (64, 64, 64, 3)$  voxel grids  $V_{64,64,64,3}$ . The deviation features  $d$  include deviations in all directions for all points  $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ . The shape error after the final station ( $x^{N_s,4}$ ) is used as input while the process parameters  $y$  and upstream stations shape errors  $x^{1,4}, \dots, x^{N_s-1,4}$  are used as output. The model architecture hyperparameters were selected as proposed in the OSER-MAS approach. Training hyperparameters were optimized for all scenarios. Adam optimizer [60] is used for training in scenarios (i), (ii) and (iii). Initial learning rates  $\alpha_0 = [0.1, 0.01, 0.001, 0.0001]$  were compared for scenario (i) and (ii),  $\alpha_0 = 0.001$  gave optimal performance in terms of error and convergence,  $\alpha_0 = [0.1, 0.01]$  gave inferior performance as compared to  $\alpha_0 = [0.001, 0.0001]$ ,  $\alpha_0 = 0.001$  was finally selected as the learning rate gives faster convergence between the two values. The same combinations were tested

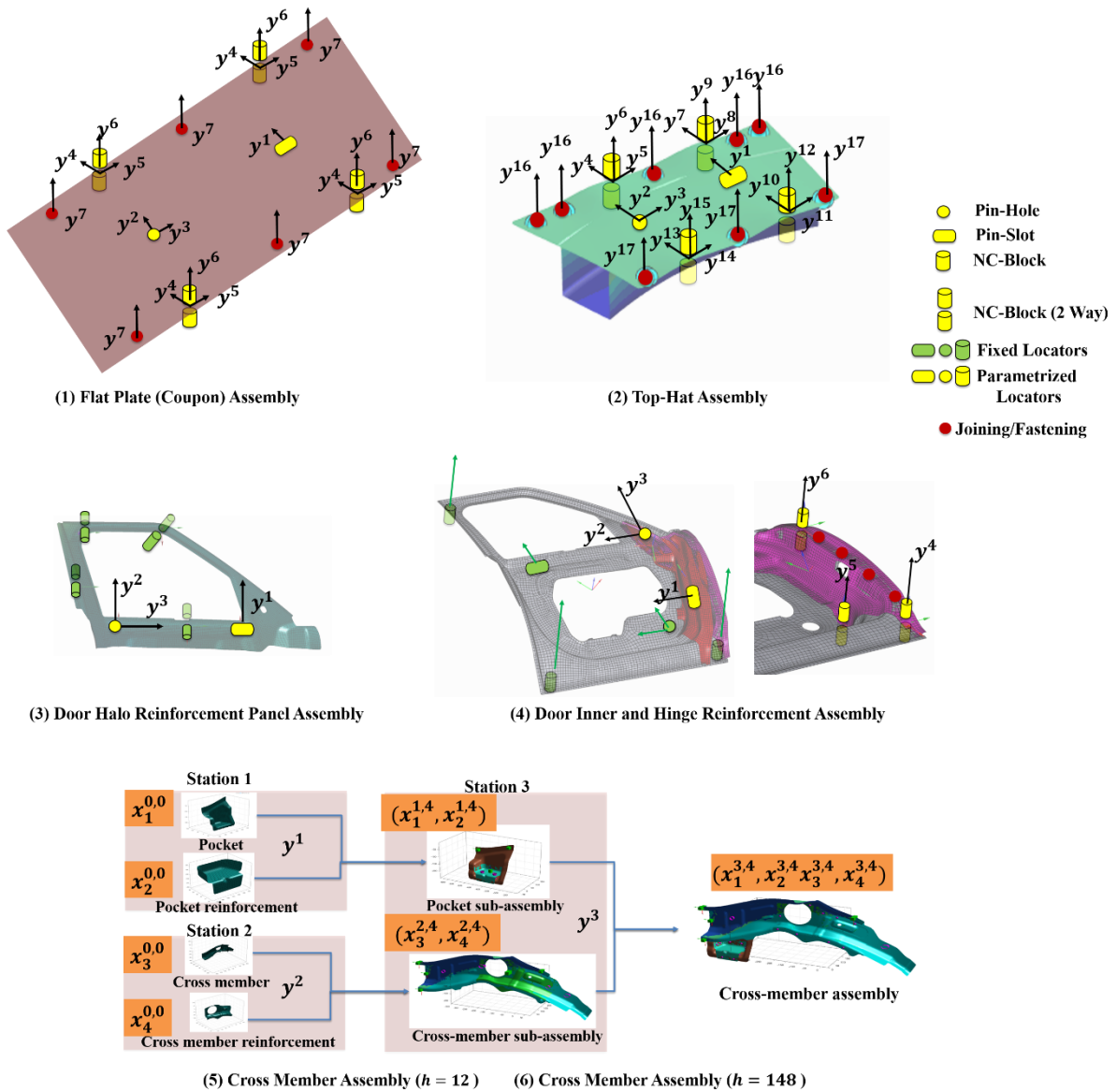


FIGURE 4. Assembly Cases

for scenario (iii) while under the constraint that  $\alpha_C = \alpha_F/10$  given the fine-tuning protocol to ensure later layers learn at a faster rate as compared to initial layers.  $\alpha_C = 0.0001$  and  $\alpha_F = 0.001$  gave the most optimal performance. Scenario (iv) tested initial learning rate for stochastic gradient descent (SGD),  $\alpha_0 = 0.001$  gave optimal performance. The learning rates were multiplied in each case by the weight uncertainty as described in Table II. Minibatch sizes of 8, 16 and 32 were tested. Larger batch sizes could not be used given the high GPU memory requirements of 3D CNNs. Minibatch size of 32 gave the best performance. Smaller sizes such as 8 and 16 caused the training process to be unstable. The model is trained for 300 epochs. Group normalization [61] with four groups is used after each convolutional layer. This prevents overfitting and accounts for small minibatch size due to GPU memory size constraints and aids in stabilizing the training

process. Optimization for training hyperparameters was done for case (1) (Flat Plate Assembly) to ensure computation feasibility. Given case studies (1) to (4) consist of a single station; the decoder of the 3D U-Net model is not used as upstream shape error for previous station stations does not

TABLE IV  
ASSEMBLY CASES

Case	Geometry	Input Parts	Stages	Stations	Process Parameters
(1) Flat Plate	2D	1 ideal part	4	1	7
(2) Top-Hat	3D	2 ideal parts	4	1	17
(3) Door Halo	3D	1 ideal parts	2	1	3
(4) Door Inner and Hinge	3D	2 ideal parts	4	1	6

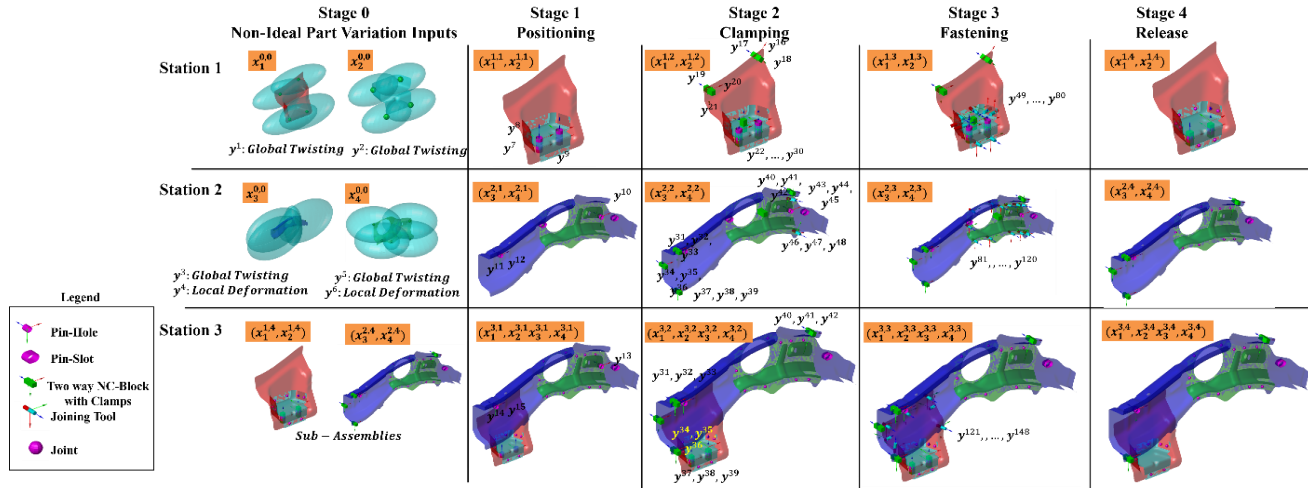


FIGURE 5. Process Parameters for Cross Member Assembly for case (5) and (6)

(5) Cross Member	3D	4 non-ideal parts	12	3	12
(6) Cross Member	3D	4 non-ideal parts	12	3	148

need to be estimated. Case study (5) leverages the decoder to estimate shape error after upstream stations. The work has been implemented using Python 3.7 and TensorFlow - GPU 2.1 [62] and TensorFlow Probability 0.9. A python library named DLMFG [63] has been developed to validate and replicate the results of the methodology. For this paper, both, the data generation and evaluation of the approaches have been done using VRM. Two Nvidia Tesla V100 32 GB GPUs are used for model training and deployment.

## B. DISCUSSION: SCALABILITY

The results for training assembly cases sequentially in the aforementioned scenario are summarized in Fig. 6. Training in all scenarios is done until convergence. Model is considered converged when the performance metrics on the test set  $E = (X^E, Y^E)$  are better than the threshold. R-Squared ( $R^2$ )  $\geq 0.90$  is considered as the convergence criteria for  $y^r$  and Receiving Operating Characteristics – Area Under Curve (ROC-AUC)  $\geq 0.90$  is considered as a convergence for  $y^c$ . The initial training set size  $T^0$  is set to be 500 samples and  $T^i = 100$  samples are added in each closed-loop iteration based on the estimated GMM parameters. Based on empirical tests the pre-specified number of mixtures in the GMM is fixed at  $K = 5$ .

The results show that for low complexity cases such as Flat Plate (1) and Top-Hat (2) the effects of using closed-loop sampling with continual or transfer learning gives only minor reductions in training samples for converging. As the complexity of the assembly cases increases and the effect of pre-trained weights of continual and transfer learning become significant reduction up to 50% in the number of required training samples for case (6) can be seen. This

validates the need for scalable approaches required for training high-dimensional assembly cases while leveraging the pre-trained models on low-dimensional assembly cases. The performance measure of using continual learning for all  $T_n$  tasks are done by comparing  $R^2$  on the  $T_i$ th task when learning is performed only till  $T_i$ th task ( $R_{T_i, T_i}^2$ ) and when learning is performed till the  $T_n$ th task ( $R_{T_i, T_n}^2$ ). Catastrophic forgetting for each task ( $CF_{T_i}$ ) is quantified as the difference between performance in the aforementioned situations:

$$CF_{T_i} = R_{T_i, T_n}^2 - R_{T_i, T_i}^2 \quad (19)$$

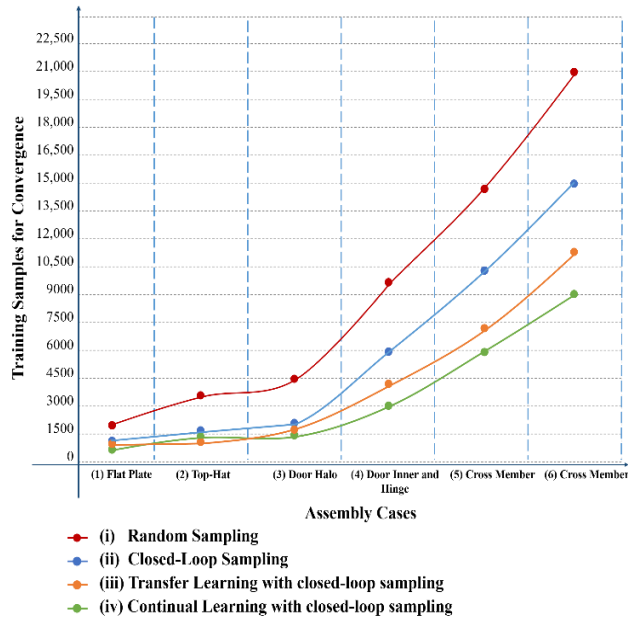
Negative value of  $CF_{T_i}$  means catastrophic forgetting of previous cases while positive values mean that learning new tasks has improved the performance of previous tasks. Table V summarizes the results. Average Generalized Performance  $\frac{1}{T_n} \sum_{T_i} R_{T_i, T_n}^2$  and Average Catastrophic Forgetting  $\frac{1}{T_n} \sum_{T_i} CF_{T_i}$  are summarized and highlight generalized performance (95%) greater than the required threshold while with average forgetting of only up to 2.1%. The convergence is measured by the number of samples ( $S^{T_s}$ ) for training for the given training scenario  $T_s$ . Improvement in convergence  $\Delta_{T_s}$  for a training scenario  $T_s$  is measured as the percentage difference in training samples required for convergence as compared to the baseline training scenario which for this case is (i) random sampling

$$\Delta_{T_s} = \frac{S^{T_s} - S^{T_{s=(i)}}}{S^{T_{s=(i)}}} \quad (20)$$

Table V also summarizes the improvement in convergence for the CLIP framework i.e. improvement in convergence when using training scenario (iv) Continual Learning with



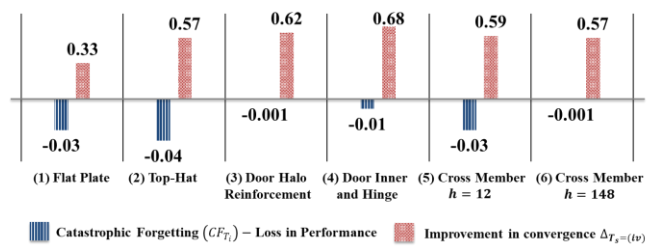
Closed-Loop Sampling  $\Delta_{T_s=(iv)}$ . Fig. 7 aims to compare loss in performance ( $CF_{T_i}$ ) with improvement in convergence ( $\Delta_{T_s=(iv)}$ ). Overall, across the six cases, the proposed CLIP framework provides 56% improvement scalability as quantified by improvement in convergence with a loss in performance of only 2.1% as quantified by Catastrophic Forgetting.



**FIGURE 6.** Convergence comparison of all assembly cases in all training scenarios

TABLE V  
CLIP FRAMEWORK PERFORMANCE

Assembly Case	$R^2_{T_i, T_n}$		$R^2_{T_i, T_i}$		$CF_{T_i}$	$\Delta_{T_s=(iv)}$
	Mean	SD	Mean	SD		
(1) Flat Plate	0.93	0.01	0.96	0.02	-0.03	0.33
(2) Top-Hat	0.91	0.02	0.95	0.01	-0.04	0.57
(3) Door Halo	0.98	0.01	0.98	0.01	0.00	0.62
Reinforcement						
(4) Door Inner and Hinge	0.97	0.01	0.98	0.02	-0.01	0.68
(5) Cross Member	0.95	0.02	0.98	0.01	-0.03	0.59
(6) Cross Member	0.96	0.03	0.96	0.03	0.00	0.57
Average Generalized Performance = <b>0.95</b>						
Average Catastrophic Forgetting = <b>-0.02</b>						
Average Convergence Improvement = <b>0.56</b>						



**FIGURE 7.** Loss in performance vs improvement in convergence comparison for CLIP

### C. DISCUSSION: INTERPRETABILITY

The interpretability of the Bayesian 3D U-Net is done on two levels:

(I) By linking requirements of the MASs with functional elements of the Bayesian 3D U-Net architecture. These include:

(1a) *Object Shape Error Voxelization*: Shape error voxelization provides an intermediate 3D data structure linking mesh obtained from CAE simulation and point clouds obtained from 3D optical scanners. Voxelization ensures that both these data structures are converted to voxels and are hence, compatible with 3D convolution operations fulfilling: Requirement (i) high data dimensionality; and, Requirement (vi) dual data generation capabilities. The voxels are multi-channels with each channel corresponding to one component of shape error. The resolution of voxels depends on the required performance. Fig. 8 shows voxelization for one component of shape error at different resolutions. Low-resolution voxels capture global shape error patterns, as the resolution is increased, local shape error patterns are effectively captured. This also increases the discriminative capability required to differentiate between collinear shape error patterns, although this comes at a higher computational cost. Empirical studies have shown that there is no significant increase in performance above  $(64 \times 64 \times 64)$  for RCA of MAS. Case (6) Cross Member Assembly ( $h = 148$ ) is used for a sensitivity study. Object Shape Error Reconstruction Error and performance is compared against voxel granularity. The reconstruction error is less than 1% given  $(64 \times 64 \times 64)$  or more granular voxels. The model performance does not increase over  $R^2 = 0.96$  even with voxels as granular as  $(96 \times 96 \times 96)$ . Additionally, with increased voxel sizes above  $(64 \times 64 \times 64)$  the minibatch size used during training has to be further reduced to 16 given GPU memory constraints which result in unstable model training and hence negatively impacts performance. Table VI summarises the results of the sensitivity study.

(1b) *Encoder with Down Sampling Kernels*: As described earlier the Bayesian 3D U-Net architecture consists of four levels of the encoder and decoder models (Fig. 2). Each level of encoding consists of the down-sampling kernel (see Down-sampling kernel in Fig. 2). The kernel consists of 3D Max pooling, which is duplicated to a residual connection and encoding connections. The residual connection consists of a 3D convolution with a filter size of one and a stride length of one in all three dimensions. The encoding connection consists of two 3D convolutions of filter size three and stride length one with ReLU activation in between. Then, the residual connection and the encoding connection are merged using element-wise addition. Finally, ReLU is applied before duplicating the output into the decoder input and next level encoder input. Overall, the down-sampling kernels in the encoder with consecutive 3D convolutions and pooling are essential for spatial correlation filtering, feature

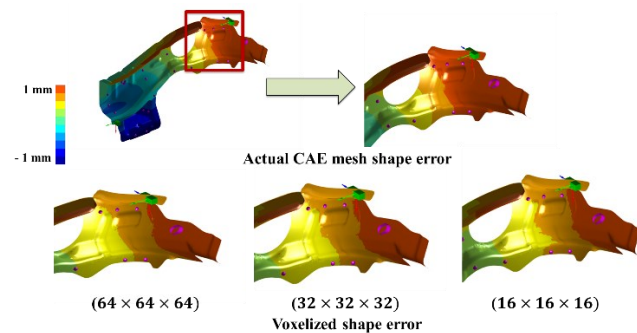


FIGURE 8. Object Shape Error Voxelization

TABLE VI  
VOXEL RESOLUTION SENSITIVITY STUDY

Voxel Resolution	Reconstruction Error	Performance ( $R^2$ )		Minibatch Size
		Mean	SD	
(16 × 16 × 16)	9.2%	0.85	0.01	64
(32 × 32 × 32)	3.8%	0.91	0.01	32
(64 × 64 × 64)	1.4%	0.96	0.02	32
(80 × 80 × 80)	1.2%	0.96	0.04	16
(96 × 96 × 96)	0.9%	0.95	0.05	16

extraction, and non-linear transformations. Consecutive levels of the decoder extract more discriminating features from the high-resolution voxelized shape error input. The discriminative ability of the features increases at each consecutive encoder level thus, enabling accurate estimate of process parameters hence high root cause isolability. Each level of the encoder is also linked to the corresponding decoder, this enables transfer of features related to the part geometry and hence enables accurate estimation of upstream part shape error at the end of the decoder. Fig. 9 highlights the features extraction capabilities of different levels of the encoder, while lower levels focus on the entire part higher levels focus on the regions that contain the shape error. This enables fulfilment of requirements: (ii) non-linearity; (iii)

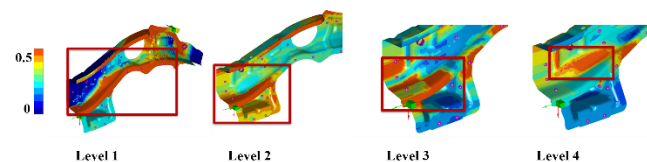


FIGURE 9. 3D Grad-CAMs for various encoder levels

collinearities; and, (iv) high faults multiplicity. The 3D Grad-CAMs for encoder levels provides interpretability by visualizing the extracted shape error features. The transparency provided in the extracted shape error features enables interpretability on why a particular root cause was isolated.

(1c) *Decoder with Up-sampling Kernels*: Each level of the decoder consists of the up-sampling kernel (see Up-sampling kernel in Fig. 2) and provides real-valued segmentation maps that estimate object shape error, i.e., the three components of

deviation for each subassembly at the end of all upstream stations. Each level of the decoder consists of two input sources; the encoder input from the corresponding level encoder and the decoder input from the previous decoder level. The following operations are then performed: (i) up-sampling of the decoder input, which is duplicated and sent to the attention gate and feature concatenation layer; (ii) the attention gate [64] distills information from the encoder and then generates relevant features that are concatenated with the up-sampling output from (i); and, (iii) this concatenated feature set is duplicated to the residual connection and the decoder connection and similar operations as in the encoder layer are performed. The number of channels of the decoder output equal to the number of components of shape error multiplied by the number of upstream stations, while the granularity of the output is the same as the input voxel size. Various levels of the decoder aggregate features from the corresponding encoder and previous decoder. This integrates part geometry features (as provided by the encoder) with the shape error features (as provided by the previous decoder) enabling accurate estimation of upstream part shape error. Different levels of decoder reconstruct shape error within different regions of the part. This enables fulfilment of requirement (viii) Fault Localization. Fig. 10(a) highlights the up-sampling capabilities of different levels of the decoder that enable estimation of object shape error of upstream assemblies. Level 2 reconstructs features of the pocket reinforcement subassembly while levels 3 and 4 reconstruct features of the cross-member reinforcement assembly. The 3D Grad-CAMs for decoder levels provides interpretability into the stations within which the fault is localized. Fig. 10(b) compares the actual upstream assembly shape errors as estimated using CAE simulation with those estimated by the decoder output.

(1d) *Multiple Output Heads*: The model consists of two output heads one head estimates real-valued process parameters  $y^r$  as done in a regression setting while the second head estimates categorical/binary process parameters  $y^c$  as done in a multi-label classification setting. This is essential for MASs as they have a large number of process parameters inclusive of (a) real-valued parameters for non-ideal parts and fixturing/tooling (within Positioning (P) and Clamping (C) stages of the PCFR assembly cycle); and, (b) binary parameters for joining operations (within the

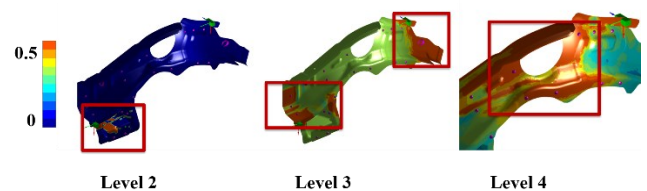


FIGURE 10(a). 3D Grad-CAMs for various levels of the decoder

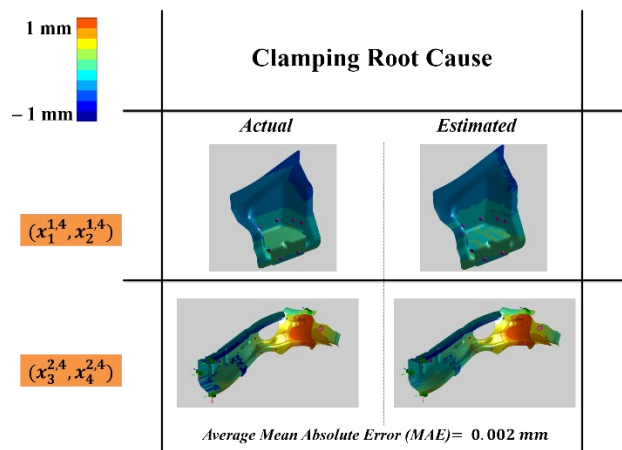


FIGURE 10(b). Comparison of actual vs estimated upstream assemblies

Fastening (F) stage of the PCFR assembly cycle). The number of regression output nodes is equal to the number of real-valued parameters and the number of classification output nodes is equal to the number of binary process

parameters. This enables fulfilment of requirement (vii) High Dimensionality and heterogeneity of process parameters.

(1e) *Attention Gate*: The soft-attention mechanism as proposed by Oktay et al. [64] is used between corresponding levels of the encoder and decoder. The attention approach allows the model to be specific to local regions. In the context of shape error estimation of assemblies, this helps the model focus on particular parts/subassemblies in each station. Adding of the attention gate increases in accuracy of upstream stations shape error estimation as the model learns where to look within the final assembly to estimate upstream sub-assemblies  $[x^{1,4}, \dots, x^{N_s-1,4}]$ . This decoder inclusive of attention gates improves performance for requirement (viii) Fault Localization. Fig. 11 shows 3D Grad-CAMs for areas of focus at different functions within the up-sampling kernel. Attention 3D Grad-CAMs enables interpretability by providing insights into the regions focused by the decoder in estimating upstream shape errors.

(1f) *Bayesian Flipout Layers*: Given the uncertainties in the system and the availability of only a limited dataset, a deterministic estimate of function  $f(\cdot)$  as shown in (2) is not

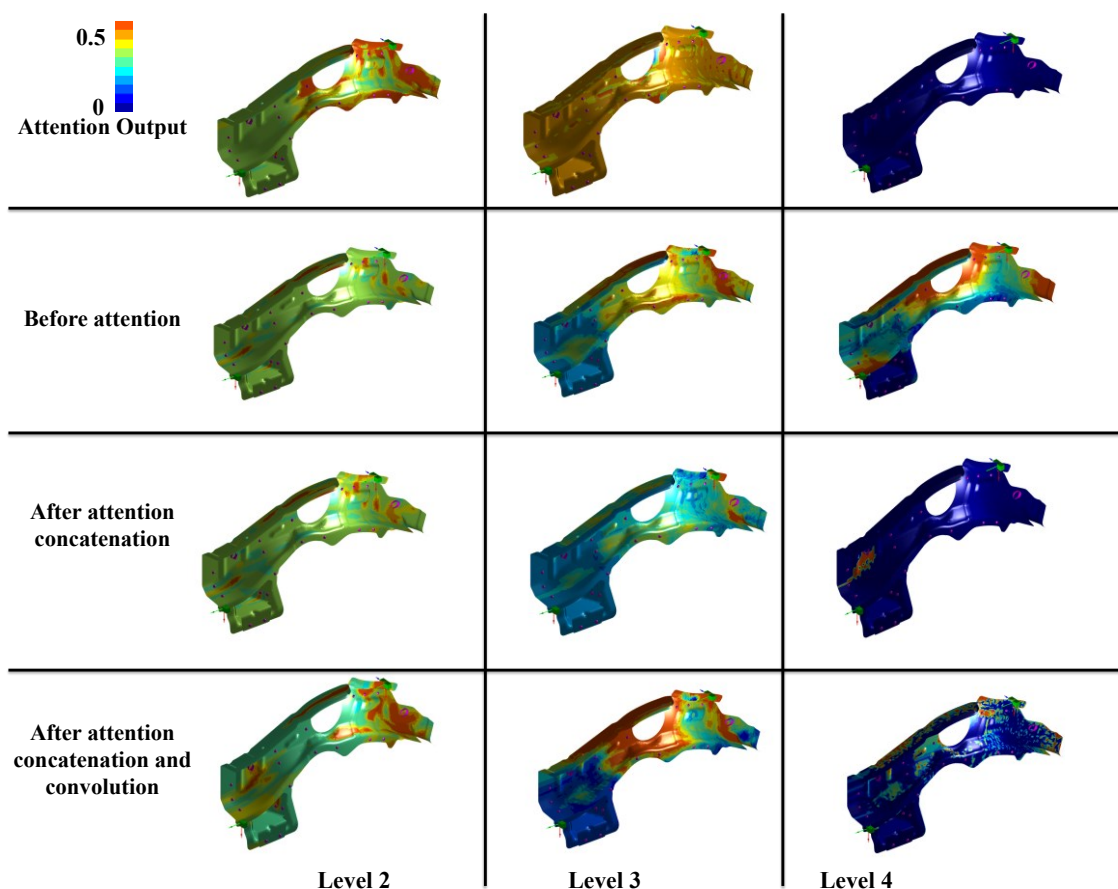


FIGURE 11. 3D Grad-CAMs for Attention at various levels of the decoder

feasible. The Flipout [65] layers leveraged in the encoder enable uncertainty quantification. These estimates of uncertainty integrate measures of confidence within isolated RC(s) and hence drive costly corrective actions [6]. This is realized by using Bayes-by-Backprop [66] which integrates backpropagation with variational inference [67] to estimate a posterior distribution  $q_{\theta}(\omega)$  which is parametrized by  $\theta$  over the neural network weights based on the pre-specified prior  $p(\omega)$ . This enables fulfillment of requirement (v) uncertainty quantification. The uncertainties are key elements of interpretability insights as they integrate a measure of confidence within the root cause estimates.

(1g) *Residual Connections*: Given the deep architecture of the model, vanishing gradients can be a major issue, hence residual [68] or skip connections are added within each down-sampling and up-sampling kernel that ensure effective prorogation of gradients by providing a skip route. Fig. 12 highlights the 3D Grad-CAMs for various stages of the residual connection. As seen (highlighted in red rectangle) in level three the layer before the residual connection has negligible activations due to skipping of the layer while the gradients become significant after the addition of the residual. The residual connections improve performance for requirements: (ii) non-linearity, (iii) collinearities, and (iv) high faults multiplicity.

(2) Using 3D Grad-CAMs to interpret the working of the architecture for different process parameter variations or root cause(s). The above 3D Grad-CAMs provide a *global* level of interpretability by linking functions elements of the architecture with requirements of the MAS. The next local

level of interpretability aims to provide transparency into the 3D Grad-CAMs for various levels of the encoder for key root cause scenarios. This links the shape error features extracted by each level of the encoder to estimate that particular root cause. Fundamentally, to interpret that the architecture is isolating a root cause correctly the features extracted by various levels of the encoder should correspond to the shape error patterns caused by that root cause. To validate this the cross member assembly (case (5)) is considered and the working of the architecture is analyzed for five key root cause(s) scenarios:

(2a) *Part Variation Root Cause*: This is caused due to variation in upstream fabrication processes is estimated as variation in  $y^m = y^1 = 2 \text{ mm}$ . Fig. 13 represents the output of the assembly given the incoming part (cross-member)  $n = 3$  has part variation. The region marked in red depicts a bend in the part that is unique to a part variation root cause [59]. The 3D Grad-CAMs as shown in Fig. 13 highlights, that the first encoder focuses around the entire part, the second encoder level can identify the edges near the bend and the final encoder levels (three and four) can identify the region where the bend has occurred and hence accurately estimate  $y^1$  as a part variation root cause.

(2b) *Positioning Root Cause*: This is caused by tooling installation and calibration error, or tooling deterioration due to gradual wearing out of fixture locators and is estimated as variation in  $y^m = y^5 = 1 \text{ mm}$ . They affect the part placement including orientation/reorientation and stability. The 3D Grad-CAMs as shown in Fig. 14 highlights that the encoder focuses around the entire part that has an error in

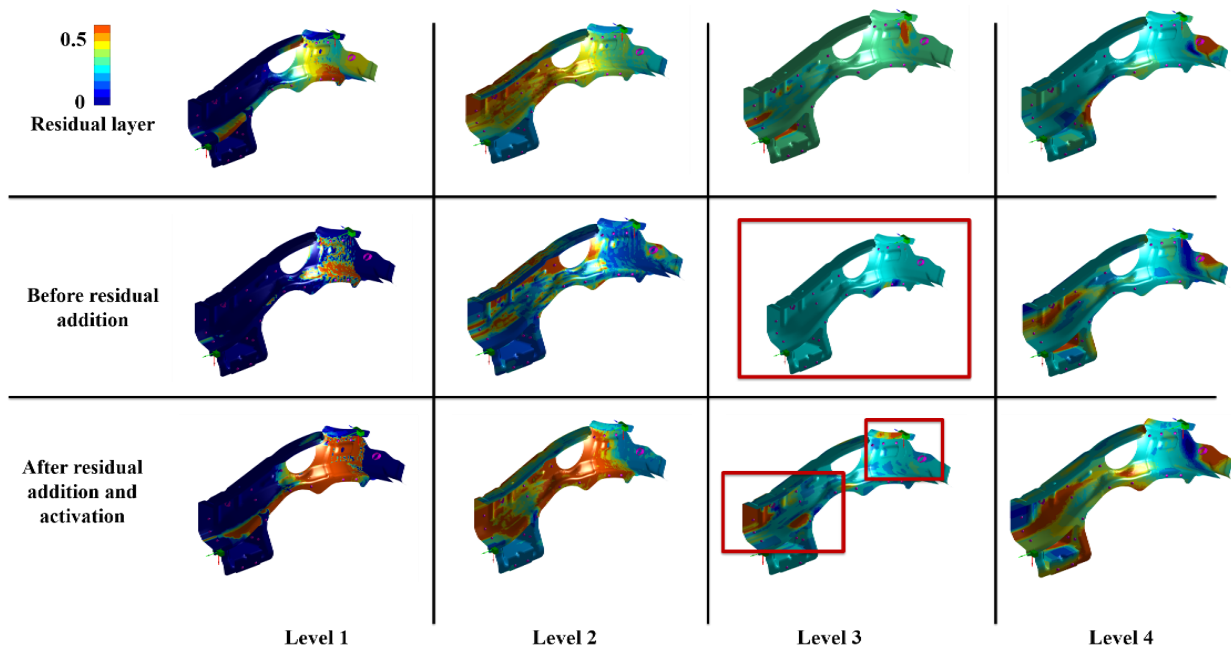


FIGURE 12. 3D Grad-CAMs for residual connection at various levels of the encoder



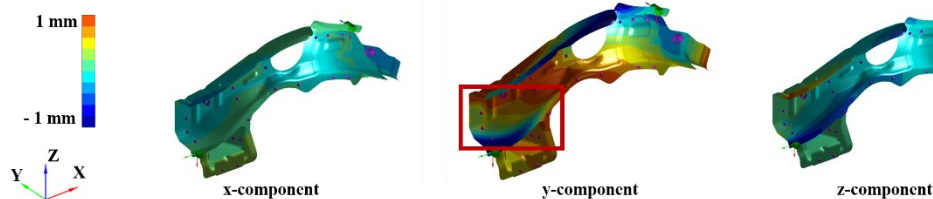


FIGURE 13(a). Part Variation Root Cause: Input Shape Error Components

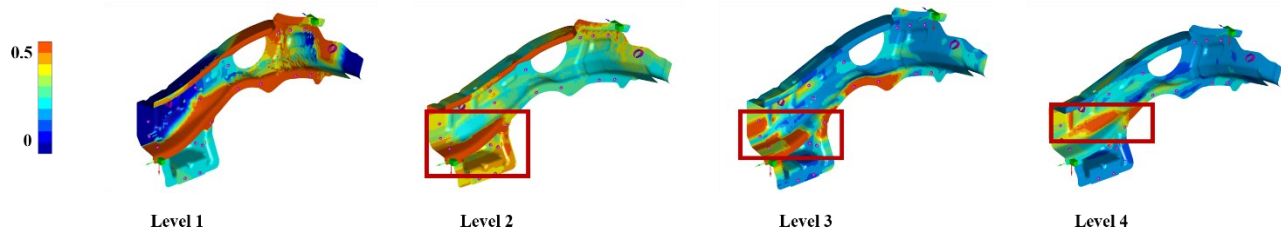


FIGURE 13(b). Part Variation Root Cause: Encoder 3D Grad CAMs

FIGURE 13. Part Variation Root Cause

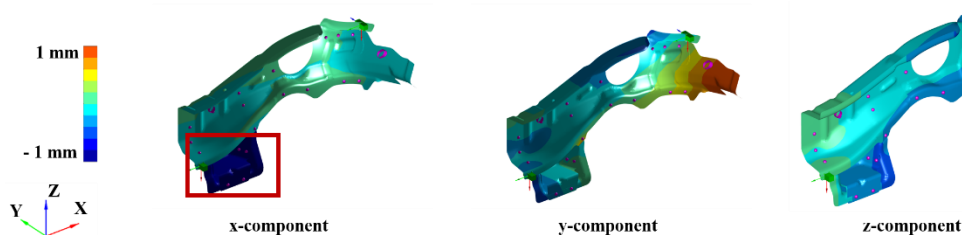


FIGURE 14(a). Pin Slot Displacement Root Cause: Input Shape Error Components

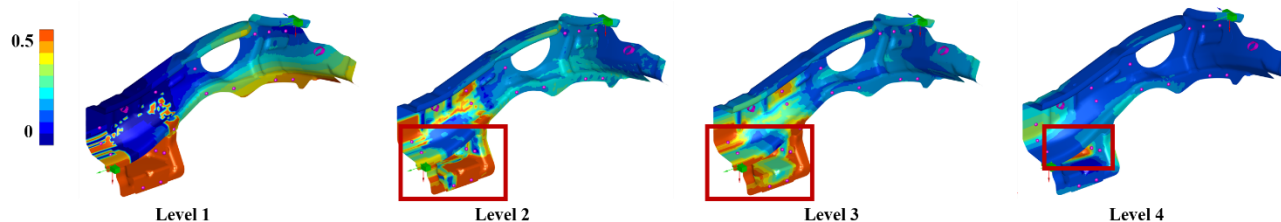


FIGURE 14(b). Pin Slot Displacement Root Cause: Encoder 3D Grad CAMs

FIGURE 14. Positioning Root Cause

orientation and estimates  $y^5$  as the magnitude of the error.

(2c) *Clamping Root Cause*: This is caused by misalignment of the clamp in the y-direction and estimated as  $y^m = y^{11} = 2 \text{ mm}$ . They cause part bending of compliant parts. The 3D grad-CAMs as shown in Fig. 15 highlights that the encoder can focus on the local bend pattern at the location of the clamp and estimate  $y^{11}$  as the magnitude of the clamp misalignment in the y-direction.

(2d) *Joining Root Cause*: This is caused by misalignment of the joining tool (SPR) in the y-direction and estimated as

$y^m = y^{12} = 2 \text{ mm}$ . They lead to a defective joint between the two assemblies. The 3D grad-CAMs as shown in Fig. 16 highlights that the first level of the encoder can focus on the region of defective joint and the later levels focus on the subassembly affected due to the defective joint and hence estimates  $y^{12}$  as the magnitude of joining tool misalignment in the y-direction.

(2e) *Part Variation and Clamping Root Causes*: This is caused when there is an upstream part variation ( $y^m = y^1 = 2 \text{ mm}$ ) and misalignment of the clamp in the y-direction

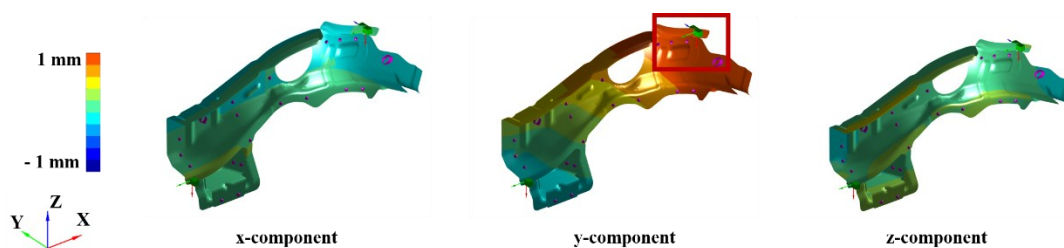


FIGURE 15(a). Clamping Root Cause: Input Shape Error Components

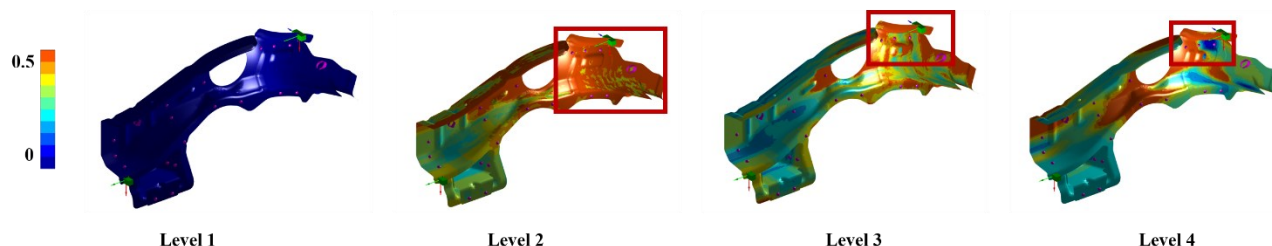


FIGURE 15(b). Clamping Root Cause: Encoder 3D Grad CAMs

FIGURE 15. Clamping Root Cause

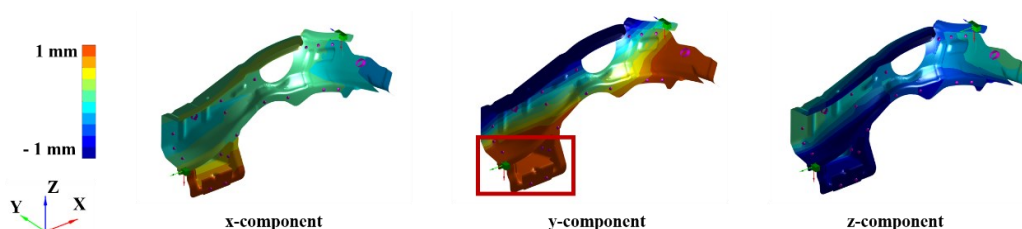


FIGURE 16(a). Joining Root Cause: Input Shape Error Components

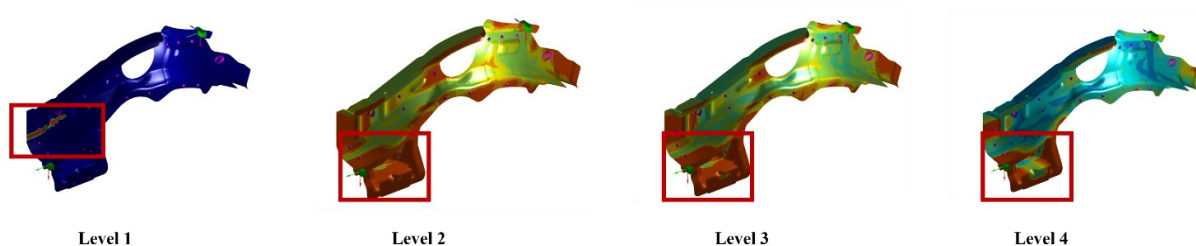


FIGURE 16(b). Joining Root Cause: Encoder 3D Grad CAMs

FIGURE 16. Joining Root Cause

( $y^m = y^{11} = 2mm$ ). These lead to multiple simultaneous bends across the assembly. The 3D Grad-CAMs as shown in Fig. 17 highlights that various levels of decoder focus on all effected regions within the sub-assembly to simultaneously estimate multiple root causes. This capability is crucial in ensuring that deep learning models have high RCA capabilities even in scenarios when all process parameters have variation and potentially are at fault (100% fault multiplicity). Such cases of high fault multiplicity cause various shape errors that are collinear (highly similar). The

ability of architecture to simultaneously focus on multiple areas within the multi-channel voxelized input and localize various bends, twists and other shape error patterns which are potentially overlapping (interacting) and then relate them to the process parameter(s) causing it, makes it the ideal approach to do RCA of high-dimensional MASs with high fault multiplicity using granular 3D data structures such as mesh (CAE) or point clouds from 3D scanners.

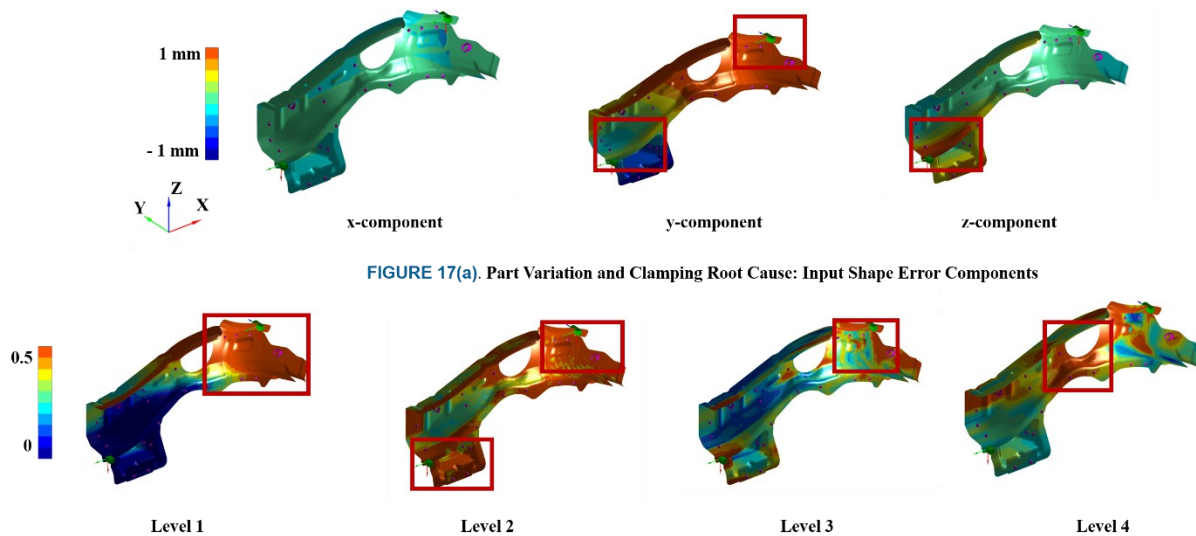


FIGURE 17(a). Part Variation and Clamping Root Cause: Input Shape Error Components

FIGURE 17(b). Part Variation and Clamping Root Cause: Encoder 3D Grad CAMs

FIGURE 17. Part Variation and Clamping Root Causes

## VI. CONCLUSIONS & FUTURE WORK

The paper proposed a novel *closed-loop in-process (CLIP)* diagnostic framework underpinned algorithm portfolio to address the current limitations of scalability and interpretability. Scalability is enabled by leveraging closed-loop training integrated with uncertainty guided continual learning or transfer learning. The approach enables effective transfer of knowledge through invariant features between MASs thereby, achieving quicker convergence with 56% lesser training samples. The overall loss in performance was limited to only 2.1 % as quantified by average catastrophic forgetting (Table V). Interpretability is enabled by leveraging 3D Grad-CAMs that provide insights into the functioning of key elements within the architecture and also relate features extracted by the architecture to shape error features within MAS. The visual interpretability explanations and uncertainty estimates integrate confidence hence, enabling *trust in black-box* deep learning models.

Scalability and interpretability are key challenges that must be solved to enable widespread adoption of deep learning methodologies in industrial environments. Key industrial application entails RCA of assembly processes of discrete components made of sheet metal parts used in automotive, aerospace or consumer products industries. These applications will leverage directly the CLIP diagnostic framework with the OSER approaches to enable scalable and interpretable root cause analysis and will be especially beneficial for processes with larger number of parts and/or larger number of assembly stations. The framework can also be leveraged for transfer of learning to different type of manufacturing processes such as stamping, machining and additive manufacturing that can be formulated using the

proposed formulation of object shape error estimation for root cause analysis, this will lead to leveraging transfer and continual learning to other manufacturing processes that can be linked to assembly processes. Interpretability has been a major barrier preventing the adoption and deployment of deep learning models in the industry. The interpretability elements proposed by the work aim to eliminate the barrier and integrate context and confidence to the estimates hence, enabling wider adoption. Leveraging such automated and interpretable RCA models provides a transformative framework by ensuring early estimation and elimination of process variations before they become defects thereby, helping to achieve Zero-Defect-Manufacturing and Right-First-Time.

Future work involves addressing the current limitations of the approach such as estimating dynamic changes in manufacturing systems and quantifying that as concept drifts or covariate shifts. Such changes when detected would then result in the model being fine-tuned such that the model accounts for the dynamic changes in the manufacturing system. This would enable lifelong learning for dynamic manufacturing environments. Further work also involves quantitative modeling of invariant features between different MASs. These invariant features can be linked to first principle models of the MASs and hence, further enhance scalability and interpretability.

## ACKNOWLEDGMENT

This study was supported by the UK EPSRC project EP/K019368/1: "Self-Resilient Reconfigurable Assembly Systems with In-process Quality Improvement" and the WMG-IIT scholarship.

## REFERENCES

- [1] P. Franciosa, M. Sokolov, S. Sinha, T. Sun, and D. Ceglarek, "Deep learning enhanced digital twin for remote laser welding of aluminium structures," *CIRP Ann.*, vol. 69, no. 1, pp. 369–372, 2020.
- [2] Q. Qi and F. Tao, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
- [3] W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.
- [4] V. J. Shahi, A. Masoumi, P. Franciosa, and D. Ceglarek, "A quality-driven assembly sequence planning and line configuration selection for non-ideal compliant structures assemblies," *Int. J. Adv. Manuf. Technol.*, vol. 106, no. 1–2, pp. 15–30, 2020.
- [5] S. Sinha, P. Franciosa, and D. Ceglarek, "Object Shape Error Response using Bayesian 3D Convolutional Neural Networks for Assembly Systems with Compliant Parts," *IEEE Trans. Ind. Informatics*, 2020.
- [6] S. Sinha, P. Franciosa, and D. Ceglarek, "Object Shape Error Response using Bayesian 3D U-Net for Multi-Station Assembly Systems with Non-Ideal Compliant Parts," *IEEE Trans. Ind. Informatics*, 2020.
- [7] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning," *IEEE Trans. Ind. Informatics*, vol. 15, no. 4, pp. 2446–2455, 2019.
- [8] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided Continual Learning with Bayesian Neural Networks," *arXiv:1906.02425*, 2019.
- [9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- [10] G. Putnik et al., "Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap," *CIRP Ann. - Manuf. Technol.*, vol. 62, no. 2, pp. 751–774, 2013.
- [11] P. Wang and R. X. Gao, "Transfer learning for enhanced machine fault diagnosis in manufacturing," *CIRP Ann.*, vol. 69, no. 1, pp. 413–416, Jan. 2020.
- [12] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, pp. 19990–19999, 2019.
- [13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.
- [14] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, 2012.
- [15] H. C. Shin et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [16] R. Zhang, H. Tao, L. Wu, and Y. Guan, "Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions," *IEEE Access*, vol. 5, pp. 14347–14357, 2017.
- [17] J. Wang and C. Zhao, "Mode-cloud data analytics based transfer learning for soft sensor of manufacturing industry with incremental learning ability," *Control Eng. Pract.*, vol. 98, p. 104392, 2020.
- [18] F. Feng, R. H. M. Chan, X. Shi, Y. Zhang, and Q. She, "Challenges in Task Incremental Learning for Assistive Robotics," *IEEE Access*, vol. 8, pp. 3434–3441, 2020.
- [19] G. A. Tahir and C. K. Loo, "An open-ended continual learning for food recognition using class incremental extreme learning machines," *IEEE Access*, vol. 8, pp. 82328–82346, 2020.
- [20] H. J. Song and S. B. Park, "Enriching translation-based knowledge graph embeddings through continual learning," *IEEE Access*, vol. 6, pp. 60489–60497, 2018.
- [21] A. Bielski and T. Trzcinski, "Understanding Multimodal Popularity Prediction of Social Media Videos With Self-Attention," *IEEE Access*, vol. 6, pp. 74277–74287, 2018.
- [22] H. Y. Chen and C. H. Lee, "Vibration Signals Analysis by Explainable Artificial Intelligence (XAI) Approach: Application on Bearing Faults Diagnosis," *IEEE Access*, vol. 8, pp. 134246–134256, 2020.
- [23] L. Zhao, Y. Zeng, P. Liu, and G. He, "Band selection via explanations from convolutional neural networks," *IEEE Access*, vol. 8, pp. 56000–56014, 2020.
- [24] D. Ceglarek and J. Shi, "Dimensional variation reduction for automotive body assembly," *Manuf. Rev.*, vol. 8, pp. 139–154, 1995.
- [25] J. Shi and S. Zhou, "Quality control and improvement for multistage systems: A survey," *IIE Trans.*, vol. 41, no. 9, pp. 744–753, Jul. 2009.
- [26] Y. Liu, R. Sun, and S. Jin, "A survey on data-driven process monitoring and diagnostic methods for variation reduction in multi-station assembly systems," *Assembly Automation*, vol. 39, no. 4., pp. 727–739, 2019.
- [27] D. W. Apley and J. Shi, "Diagnosis of Multiple Fixture Faults in Panel Assembly," *J. Manuf. Sci. Eng.*, vol. 120, no. 4, p. 793, 2008.
- [28] M. Chang and D. C. Gossard, "Computational method for diagnosis of variation-related assembly problems," *Int. J. Prod. Res.*, vol. 36, no. 11, pp. 2985–2995, 1998.
- [29] K. G. Yu, S. Jin, and X. M. Lai, "Fixture variation diagnosis of compliant assembly using sensitivity matrix," *J. Shanghai Jiaotong Univ.*, 2009.
- [30] R. Agrawal, J. F. Lawless, and R. J. Mackay, "Analysis of variation transmission in manufacturing processes - Part II," *J. Qual. Technol.*, vol. 31, no. 2, pp. 143–154, 1999.
- [31] C. Zou, W. Jiang, and F. Tsung, "A lasso-based diagnostic framework for multivariate statistical process control," *Technometrics*, vol. 53, no. 3, pp. 297–309, 2011.
- [32] Y. Shang, F. Tsung, and C. Zou, "Statistical process control for multistage processes with binary outputs," *IIE Trans. (Institute Ind. Eng.)*, vol. 45, no. 9, pp. 1008–1023, 2013.
- [33] J. Jin and J. Shi, "State Space Modeling of Sheet Metal Assembly for Dimensional Control," *J. Manuf. Sci. Eng.*, vol. 121, no. 4, p. 756–762, 1999.
- [34] Y. Ding, D. Ceglarek, and J. Shi, "Fault Diagnosis of Multistage Manufacturing Processes by Using State Space Approach," *J. Manuf. Sci. Eng.*, vol. 124, no. 2, p. 313–322, 2002.
- [35] Y. Ding, J. Shi, and D. Ceglarek, "Diagnosability Analysis of Multi-Station Manufacturing Processes," *ASME. J. Dyn. Sys., Meas., Control.*, vol. 124, no. 1, pp. 1–13, 2002.
- [36] Y. Ding, S. Zhou, and Y. Chen, "A comparison of process variation estimators for in-process dimensional measurements and control," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 127, no. 1, pp. 69–79, 2005.
- [37] D. Ceglarek and P. Prakash, "Enhanced piecewise least squares approach for diagnosis of ill-conditioned multistation assembly with compliant parts," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 226, no. 3, pp. 485–502, 2012.
- [38] D. Ceglarek and J. Shi, "Fixture Failure Diagnosis for Autobody Assembly Using Pattern Recognition," *J. Eng. Ind.*, vol. 118, no. 1, p. 55–66, 1996.
- [39] D. Ceglarek and J. Shi, "Fixture failure diagnosis for sheet metal assembly with consideration of measurement noise," *J. Manuf. Sci. Eng.*, vol. 121, no. 4, p. 771–777, 1999.
- [40] Y. G. Liu and S. J. Hu, "Assembly Fixture Fault Diagnosis Using Designated Component Analysis," *J. Manuf. Sci. Eng.*, vol. 127, no. 2, p. 358–368, 2005.
- [41] D. Ceglarek, J. Shi, and S. M. Wu, "A Knowledge-Based Diagnostic Approach for the Launch of the Auto-Body Assembly Process," *J. Eng. Ind.*, vol. 116, no. 4, p. 491–499, 1994.
- [42] S. Du and L. Xi, "Fault diagnosis in assembly processes based on engineering-driven rules and PSOSAEN algorithm," *Comput. Ind. Eng.*, vol. 60, no. 1, pp. 77–88, 2011.
- [43] S. Du, J. Lv, and L. Xi, "An integrated system for on-line intelligent monitoring and identifying process variability and its application," *Int. J. Comput. Integr. Manuf.*, vol. 23, no. 6, pp. 529–542, 2010.
- [44] G. Beruvides, A. Villalonga, P. Franciosa, D. Ceglarek, and R. E. Haber, "Fault pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts based on reinforcement learning architecture," in *Procedia CIRP*, vol. 67, pp. 601–606, 2018.
- [45] S. Dey and J. A. Stori, "A Bayesian network approach to root cause diagnosis of process variations," *Int. J. Mach. Tools Manuf.*, vol. 45, no. 1, pp. 75–91, 2005.



- [46] Y. Liu and S. Jin, "Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets," *Int. J. Adv. Manuf. Technol.*, vol. 65, no. 9–12, pp. 1229–1237, 2013.
- [47] Sinha S., Franciosa P., Ceglarek D., "Object Shape Error Response using Bayesian 3D Convolutional Neural Networks for Assembly System with Compliant Parts," in *Proc. 18th IEEE Int. Conf. on Ind. Inf.*, pp 104–109, 2020.
- [48] G. Arvanitis, A. S. Lalos, and K. Moustakas, "Robust and Fast 3D Saliency Mapping for Industrial Modeling Applications," *IEEE Trans. Ind. Informatics*, vol. 17, no. 2, pp. 1307–1317, 2020.
- [49] L. Da Xu, C. Wang, Z. Bi, and J. Yu, "AutoAssem: An automated assembly planning system for complex products," *IEEE Trans. Ind. Informatics*, vol. 8, no. 3, pp. 669–678, 2012.
- [50] Q. Rong, J. Shi, and D. Ceglarek, "Adjusted Least Squares Approach for Diagnosis of Ill-Conditioned Compliant Assemblies," *J. Manuf. Sci. Eng.*, vol. 123, no. 3, p. 453–461, 2001.
- [51] F. Adly et al., "Simplified subspace regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Ind. Informatics*, vol. 11, no. 6, pp. 1267–1276, 2015.
- [52] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems*, pp. 580–590, 2017.
- [53] X. Ding, Y. Chen, Z. Tang, and Y. Huang, "Camera Identification Based on Domain Knowledge-Driven Deep Multi-Task Learning," *IEEE Access*, vol. 7, pp. 25878–25890, 2019.
- [54] K. Mannar, D. Ceglarek, F. Niu, and B. Abifaraj, "Fault region localization: Product and process improvement based on field performance and manufacturing measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 423–439, 2006.
- [55] X. W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2. Institute of Electrical and Electronics Engineers Inc., pp. 514–525, 2014.
- [56] P. Franciosa, A. Palit, S. Gerbino, and D. Ceglarek, "A novel hybrid shell element formulation (QUAD+ and TRIA+): A benchmarking and comparative study," *Finite Elem. Anal. Des.*, vol. 166, p. 103319, Nov. 2019.
- [57] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
- [58] H. Yosinski, J. Clune, J., Bengio, Y., & Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.
- [59] W. Huang and D. Ceglarek, "Mode-based decomposition of part form error by discrete-cosine-transform with implementation to assembly and stamping system with compliant parts," *CIRP Ann. - Manuf. Technol.*, vol. 51, no. 1, pp. 21–26, 2002.
- [60] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR*, 2015.
- [61] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- [62] GoogleResearch, "TensorFlow: Large-scale machine learning on heterogeneous systems," *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pp. 265–283 2015.
- [63] S. Sinha, P. Franciosa, and D. Ceglarek, "Bayesian deep learning for manufacturing," 2020. [Online]: Available. [https://github.com/sumitsinha/Deep\\_Learning\\_for\\_Manufacturing](https://github.com/sumitsinha/Deep_Learning_for_Manufacturing)
- [64] O. Oktay et al., "Attention u-net: Learning where to look for the pancreas," *arXiv Prepr. arXiv1804.03999*, 2018.
- [65] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," in *6th International Conference on Learning Representations (ICLR)*, 2018.
- [66] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 2, pp. 1613–1622, 2015.
- [67] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *J. Am. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2016.
- [68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.



**SUMIT SINHA** received his Bachelor of Technology (B.Tech.) degree in Industrial & Systems Engineering from the Indian Institute of Technology (IIT), Kharagpur in 2017. He is currently pursuing his Ph.D. degree in deep learning for manufacturing at WMG, University of Warwick, Coventry, UK. From 2017 to 2018, he worked as a data scientist at ZS Associates in the application of data science in pharmaceutical sales and marketing. He also worked briefly on data science projects at the Hong Kong University (HKU) and Chongqing University, China. His research interests include data science, Bayesian deep learning and digital twins of manufacturing assembly systems.



**PASQUALE FRANCIOSA** received the Ph.D. degree in mechanical engineering from the University of Naples Federico II, Italy, in 2010. He is an Assistant Professor with the University of Warwick, Coventry, U.K., and the Head of the Laser Welding Applications Laboratory, Warwick Manufacturing Group, University of Warwick. He has been Principal Investigator (PI) and Co-Investigator (Co-I) on several funded projects with a total income to the University of Warwick of circa £2.8M since 2015. He has authored or coauthored more than 80 papers. His research interests include smart manufacturing, process monitoring, closed-loop control, applications of machine learning/artificial intelligence, and multidisciplinary optimization, with specific attention on assembly systems and laser processes.

Dr. Franciosa was the recipient of four Best Paper Awards. He is a Member of the Editorial Board of the *ASTM Smart and Sustainable Manufacturing Systems* journal.



**DARIUSZ CEGLAREK** received the Ph.D. degree in mechanical engineering (ME) from the University of Michigan-Ann Arbor, Ann Arbor, MI, USA, in 1994. He is an EPSRC Star Recruit Research Chair with the University of Warwick, Coventry, U.K. He was a Professor with IS&E, University of Wisconsin-Madison, Madison, WI, USA. He has been Principal Investigator (PI)/Co-Investigator (Co-I) on research grants of over £30M: NSF/ NIST/ EPSRC/ InnovateUK / APC / EU-FP7/ Curie and industry. He has authored or coauthored more than 180 papers and is listed by Stanford University among top 2% of the world's leading scientists. His research interests include smart manufacturing and data mining/AI for root cause analysis across design, manufacturing, and service.

Prof. Ceglarek is a Fellow of College International pour la Recherche en Productique (CIRP). He was the recipient of several Best Paper Awards, the 2018 JLR "Innovista" Award for the most innovative "piloted technology," the EPSRC Star Award, and the NSF CAREER Award. He was the Associate Editor for *ASTM Smart and Sustainable Manufacturing Systems*, the *IEEE Transactions on Automation Science and Engineering*, and the *ASME Journal of Manufacturing Science and Engineering*.